NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

ACTIVE VIBRATION CONTROL METHOD FOR SPACE TRUSS USING PIEZOELECTRIC ACTUATORS AND FINITE ELEMENTS

by

Carey M. Pantling

December 1999

Thesis Advisor: Young S. Shin
Co-Advisor: Brij N. Agrawal

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1999	Master's Thesis

4. TITLE AND SUBTITLE
ACTIVE VIBRATION CONTROL METHOD FOR SPACE TRUSS USING PIEZOELECTRIC ACTUATORS AND FINITE ELEMENTS

5. FUNDING NUMBERS

6. AUTHOR(S)
Pantling, Carey M.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

This thesis created an analytical model for active vibration control of the NPS space truss using ANSYS. The NPS space truss is a 3.7-meter long truss that simulates a space-borne appendage with sensitive equipment at its extremities. With the use of a dSPACE data acquisition and processing system, quartz force transducer and piezoelectric actuator, active controls using an integral plus double integral control law were used to damp out the vibrations caused by a linear proof mass actuator. Vibration reductions on the order of 15-20 dB were obtained with experiment. The ANSYS finite element model used SOLID5 elements to model the piezoelectric characteristics and ANSYS Parametric Design Language to provide for an iterative approach to an active controls analysis. Comparative data runs were performed with the ANSYS model to determine its similarity to experiment. The analytical model produced power reductions of 18-22 dB, demonstrating the ability to model the control authority with a finite element model. This technique can be used and modified to enhance its flexibility to many types of controls and vibration reduction applications. An analytical model for active control of the NPS space truss using MATLAB/Simulink was also developed as an alternative to the ANSYS model.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Active Vibration Control, Piezoceramic Actuators, ANSYS, Finite Element Method	211
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

ACTIVE VIBRATION CONTROL METHOD FOR SPACE TRUSS USING PIEZOELECTRIC ACTUATORS AND FINITE ELEMENTS

Carey M. Pantling
Lieutenant, United States Navy
B.S.A.E., University of Florida, 1993

Submitted in partial fulfillment of the
Requirements for the degrees of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING
MASTER OF SCIENCE IN ASTRONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1999

Author: Carey M. Pantling

Approved by: Young S. Shin, Thesis Advisor

B. N. Agrawal, Co-Advisor

Matthew D. Kelleher, Acting Chairman
Department of Mechanical Engineering

Gerald H. Lindsey, Chairman
Department of Astronautical Engineering

ABSTRACT

This thesis created an analytical model for active vibration control of the NPS space truss using ANSYS. The NPS space truss is a 3.7-meter long truss that simulates a space-borne appendage with sensitive equipment at its extremities. With the use of a dSPACE data acquisition and processing system, quartz force transducer and piezoelectric actuator, active controls using an integral plus double integral control law were used to damp out the vibrations caused by a linear proof mass actuator. Vibration reductions on the order of 15-20 dB were obtained with experiment.

The ANSYS finite element model used SOLID5 elements to model the piezoelectric characteristics and ANSYS Parametric Design Language to provide for an iterative approach to an active controls analysis. Comparative data runs were performed with the ANSYS model to determine its similarity to experiment. The analytical model produced power reductions of 18-22 dB, demonstrating the ability to model the control authority with a finite element model. This technique can be used and modified to enhance its flexibility to many types of controls and vibration reduction applications. An analytical model for active control of the NPS space truss using MATLAB/Simulink was also developed as an alternative to the ANSYS model.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ADC	Analog to Digital Conversion
ADCH	Analog to Digital Channel
APDL	ANSYS Parametric Design Language
CPU	Central Processing Unit
DAC	Digital to Analog Conversion
DACH	Digital to Analog Channel
DC	Direct Current
DCM	Direction Cosine Matrix
DOF	Degree of Freedom
DSA	Dynamic Spectrum Analyzer
D&D	Drag and Drop
EVA	Extra-Vehicular Activity
FBG	Fiber-optic Bragg Grating
FEM	Finite Element Method Finite Element Model
GUI	Graphic User Interface
IGain	Single Integral Gain
IIGain	Double Integral Gain
I/O	Input / Output
IDIFF	Integral Plus Double Integral Force Feedback
NASA	National Air and Space Administration
NPS	Naval Postgraduate School
NRL	Naval Research Laboratory
PPF	Positive Position Feedback
PZT	Lead Zirconate Titanate
RTI	Real Time Integration
RTILIB	RTI Library
SG	System Gain
S/N	Serial Number
SRDC	Spacecraft Research and Design Center
SSL	Smart Structures Laboratory

LIST OF SYMBOLS

A	Area
[AP]	Application Vector
[c]	Elasticity Matrix
C	Control Signal
[C]	Damping Matrix
[d]	Piezoelectric Matrix (strain)
[e]	Piezoelectric Matrix (stress)
E	Elastic Modulus
$\vec{D}$	Electric Flux Vector
$\vec{E}$	External Electric Field Vector
G	Modulus of Rigidity
G_1	Feedback Gain
[H]	Shaping Matrix
H	Transfer Function
I	Moment of Inertia
[ISO]	Isolation Vector
J	Polar Moment of Inertia
k	Elemental Stiffness
[K]	Stiffness Matrix
K	Generic Constant
L	Length
[M]	Mass Matrix
$\vec{P}$	Poling Axis
[s]	Compliance Matrix
[S]	Strain Tensor (Vector)
[T]	Stress Tensor (Vector)
U	Potential Energy
V	Voltage
x	Displacement (generic)
X	Typical Input
Y	Typical Output
z	Digital Sequence Term
α	Rayleigh Damping Constant
β	Rayleigh Damping Constant
δ	Displacement (small)
$\in$	Strain, Axial
$[\varepsilon^{\epsilon}]$	Dielectric Matrix, Constant $\in$
$[\varepsilon^{\sigma}]$	Dielectric Matrix, Constant σ
ρ	Density
σ	Stress
ζ	Damping Ratio

ACKNOWLEDGMENTS

The author would like to thank the persons and companies that assisted in the development of this thesis.

First, I wish to thank Professors Young Shin and Brij Agrawal for their support during the course of this work. Specifically, thanks to Professor Shin for driving me and opening up the alternative doors that led to the success of this thesis and to Professor Agrawal for providing the facilities and hardware used for the research.

My special thanks also to Sheldon Imaoka at Collaborative Solutions, Inc. for being available for all manner of odd questions on the inner workings of ANSYS. I would not have been able to complete this thesis without your timely assistance; for that I thank you. To Boeing Information, Space & Defense Systems, MRJ Technology Solutions, NASA/Johnson Space Center and the Naval Research Laboratory, my thanks for allowing me the time and facilities to study smart structures, and the platforms that will eventually carry them. The individuals at these organizations that I am grateful to are Robert Boucher and Joseph Dutson at Boeing, Vit Babuška at MRJ, Vic Cooley at NASA and Shalom Fisher and Barry Dunn at the NRL.

To CDRs Michael McMaster and Joe Welch, I wish to extend my personal thanks for keeping a watchful eye on me while here at the Naval Postgraduate School and for giving me good guidance for the rest of my naval career.

Finally, I wish to make specific mention of my family. To my sons, Aaron and Adam, thanks for giving me the largest grins ever when I walked in the door. Your Dada loves you both very much.

To my wife Veronica, I couldn't have done this without your continued support and love; your contribution to this work was critical to its completion. My mental and physical health would be in shambles now without your smiling face and loving care. I love you more than you know, but then again, you probably do.

I. INTRODUCTION

A. BACKGROUND

As the 21st Century approaches, the use of space and satellites will continue to grow. Even as technology yields smaller devices with reduced power requirements, the need for faster and higher-bandwidth communication has led to larger satellites. As satellites have become larger and more complex, the launch booster capability has not been appreciably increased. This has led engineers to design their satellites to be lighter and more compact, and able to deploy large lightweight antennas and reflectors when on orbit. The main structure of the International Space Station (ISS), for example, is a 360-foot long truss, when measured end to end. To meet the logistics of bringing such a large structure to orbit, weight has been a primary consideration.

These lighter, more flexible structures are prone to low frequency vibration, which brings with it new challenges for dynamic control. Accurate modeling of the on orbit characteristics of these structures is essential to being able to reduce their vibration in the space environment. When on orbit, perturbations may come from a number of sources, for example: attitude control maneuvers, crew motion, thermal effects and docking and undocking of various orbital transfer vehicles like Soyuz and the Space Shuttle. These disturbances may cause large and unwanted vibrations in the structure that may disturb sensitive operations.

In order to isolate the sensitive equipment from these vibrations, there are three possible locations to isolate the disturbance. These options are to isolate at the source of the disturbance, isolate equipment at the sink with rack-mounted isolation gear or to

remove energy along the disturbance path, usually the spacecraft structure [Ref. 1]. This thesis will discuss vibration isolation along the structural path.

Inherent in all structures is a degree of natural passive damping. This may be enhanced with installed devices, such as visco-elastic dampers. These devices may be large, and for larger structures may not be worth the additional lift capacity required to bring them to orbit. Active damping is the second option available for vibration reduction.

Implementing an active damping system can be difficult due to the difficulties in modeling the dynamic characteristics of the structure. However, once a model is developed the control laws and active control devices may then be chosen and evaluated.

The Naval Postgraduate School (NPS) Space Truss is a small-scale flexible structure that has a piezoceramic actuator installed. Previous experiments have shown that large reductions in vibration amplitude can be gained with the proper control law selection and appropriate gains. Piezoceramic actuators were selected because of their high bandwidth and low power consumption. The NPS Space Truss is now a test platform for further experiments on active vibration reduction.

By applying a sophisticated computer-modeling program, a detailed dynamic model of the NPS Space Truss can be obtained, and active controls simulated on this model. The simulated results can then be verified with an experiment involving the NPS Space Truss. The techniques learned in modeling the NPS Space Truss may then be applied to larger truss structures, such as the International Space Station. Once a model is obtained, a simulation of active controls may be performed, which may yield results that are useful for future integration into a structure's design.

B. OBJECTIVES

The objective of this thesis were as follows:

- Create a finite element model (FEM) of the NPS Space Truss.
- Develop an active vibration control model and integrate into the FEM.
- Compare the active FEM to experimental results.

C. SCOPE OF THESIS

The scope of this thesis includes:

- Integration of a new dSPACE digital signal processing system [Ref. 2] as the heart of the active controls system on the NPS space truss.
- Creation and evaluation of an actively controlled model of the NPS space truss using ANSYS [Ref. 3].
- Comparison of the FEM with the actual truss by experiment.

The use of ANSYS, MATLAB [Ref. 4] and dSPACE was instrumental to the work contained herein. This thesis is not intended to replace the use of tutorials available in MATLAB, ANSYS or dSPACE. On-line and printed guides and tutorials are available to give the un-experienced layman an opportunity to learn the system in order to use the information presented. A basic outline of how to get the programs and systems operating is included, and has sufficient detail to allow a user with some experience to run the desired programs. Where specific program instructions are given, menu commands will be used with a ">" to indicate a sub-menu selection, or the commands as typed.

D. METHODOLOGY

The research path taken for this thesis followed several parallel paths. The research conducted involved the use of active controls systems, FEM creation, fiber-optic strain gages, and the specifics of implementation on the programs used. A study of the applications of smart structures was conducted in the scope of this research.

An independent FEM of the NPS space truss was developed using MATLAB to verify the selection of the active elements that was installed previously. This FEM was also created for integration of an alternate controllable FEM.

It was decided to create an example active vibration control model in ANSYS using generic techniques to test the method and programming algorithm. This example model consisted of a single axially loaded strut with a piezoelectric sensor and actuator. Once this method of control was verified, it was integrated into a model of the NPS space truss. Simulations were conducted in order to evaluate the FEM.

A new dSPACE digital processing system was installed and tested on the NPS space truss to allow for a dedicated hardware/software package. Once the control system was installed, it was tested under several experiments to verify its performance, and for comparison with the active controlled FEM developed with ANSYS.

E. THESIS ORGANIZATION

Chapter II contains background information, theory and a summary of the work that preceded the tasks performed by the author. Chapter III is a description of the installation of the new active controls system, and its implementation, using MATLAB /

Simulink, and dSPACE. Chapter IV contains the details on the development and implementation of an actively controlled FEM of the NPS space truss using ANSYS.

Chapter V describes the experimental methodology used to verify the NPS space truss with the finite element model created in ANSYS. Finally, Chapter VI will provide conclusions and recommendations for further study in the areas researched.

II. THE NPS SPACE TRUSS

A. TRUSS DESCRIPTION

1. Background

The NPS space truss was constructed as a continuation of a series of experiments performed by the Naval Research Laboratory (NRL) in Washington, D.C. The NRL space truss was a 3.7-meter long, 12-bay squat-"t" shaped structure. It possessed two integrated piezoelectric actuators and co-located sensors. The sensors provided input to an integral plus double integral force feedback (IDIFF) controller that fed back its signal to the actuators.

Broadband testing was performed from 0 to 100 Hz to determine the operating characteristics of the NRL space truss. The engineers at NRL obtained approximately a 100-fold reduction in amplitude power due to the active control system using the controller [Ref. 5]. In cooperation with the Spacecraft Research and Design Center (SRDC) at NPS, the NRL provided the main structural components for the assembly of an exact replica of the NRL space truss at NPS for further testing.

To date, the NPS space truss has had modal testing performed, and a single active control piezoelectric strut installed and tested. Current research into the NPS space truss involves integration of fiber-optic strain gages and finite element modeling of the truss. A catastrophic flood in the adjacent laboratory of the SRDC led to two feet of mud in the laboratory containing the NPS space truss. The truss and most of the equipment was salvaged, at minimal damage, but most of the connecting cables and documentation was

destroyed. Part of the work for this thesis involved restoring the NPS space truss to fully operational capability.

2. Elements and Construction

The NPS space truss is composed of 52 aluminum nodes joined by 161 elements in a cubic 12-bay structure. The truss measures 3.67 meters in length and 0.67 meters tall. It is attached at the center to a fixed base plate. The layout of the unmodified truss is shown in Figure 1. The NPS Space Truss.

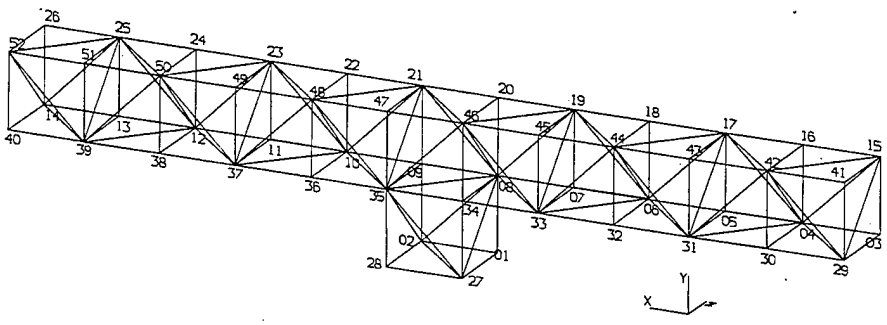


Figure 1. The NPS Space Truss

The node balls were precision milled from 7075-T6 Aluminum. They are 1.325 inches across at the threading. There are 18 threaded sites in the three axes and diagonals for flexibility in assembly. A standoff was used to provide an interface between the individual elements and the node balls. The standoff was fitted with a retaining nut, and screwed onto the node ball in the desired location.

The elements were constructed from 5/16-inch aluminum alloy tubing with 0.035-inch wall thickness. They were made in two lengths, a shorter length for the battens and longerons (100 total), and a larger length for the diagonal elements (61). A threaded sleeve was epoxied to each end of the elements to provide for mounting and precision

length adjustment. The sleeves were also pinned to the tubes for reinforcement [Ref. 6]. An exploded view of the element to node assembly is shown in Figure 2.

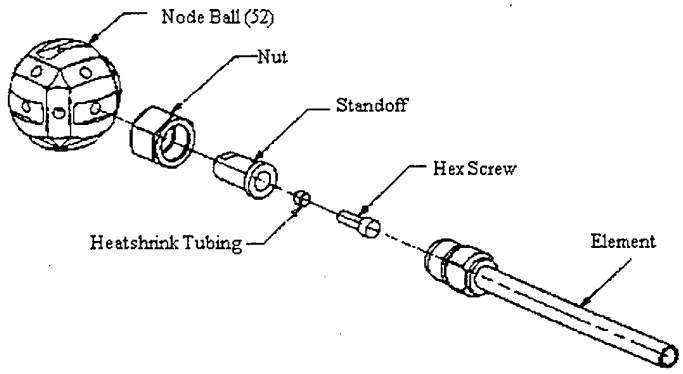


Figure 2. Node Assembly Details [From Ref. 6]

During construction of the elements, and during their assembly as a structure, great care was taken to ensure that each component was identical. To ensure this, a detailed assembly procedure was followed [Ref. 7]. The pieces were assembled such that there were no pre-stressed members in the truss. This procedure was lengthy, and took approximately two man-days to complete.

Current research at the NRL consists of a second truss of similar dimension constructed with carbon-fiber elements. The new truss was constructed without the previous rigorous torque specifications. This would better simulate the on-orbit assembly by an astronaut performing extra-vehicular activity (EVA). For an astronaut performing an EVA, some, but not all, of the assembly is rigidly specified.

After assembly, the four lower-central nodes were rigidly attached to a base plate, for mounting on a Newport Vibration Control System Table. The 1000-lb. isolation table uses compressed Nitrogen to charge pistons in its legs, providing a cushion under any mounted equipment and providing for high frequency vibration attenuation (greater than

99% above 12 Hz) [Ref. 8]. Detailed data on the NPS space truss are contained in Appendix A.

B. ACTIVE CONTROL ELEMENT

1. Overview

The active control element for the NPS space truss consists of a piezoelectric actuator and co-located force transducer. These components were mounted to steel rods in a manner to be compatible with the truss elements. Together with the computer data processing system, these components comprise the active control system.

2. Piezoelectric Theory

Piezoelectric (also called piezoceramic) materials are materials that will elongate when an electric field is applied in a pre-determined direction. Conversely, when a deformation is applied to a piezoelectric material with either an external force or strain, an internal electric potential will be developed that can be measured. In this manner the piezoelectric material can be used both a structural sensor and as an actuator.

Some piezoelectric materials, such as quartz, occur naturally, others must be synthesized [Ref. 9]. A piezoelectric material is created when a suitable material, such as Lead Zirconate Titanate (PZT) is heated above its Curie point under an externally applied electric field. The Curie point is the temperature where atomic magnetic alignment becomes unfixed. An applied electric field at this point will permanently realign the spins and magnetic dipoles [Ref. 10]. Cooling while maintaining the field will lock the

magnetic dipoles in place. This creates the poling axis, $\vec{P}$, where elongation will occur under an externally applied electric field, $\vec{E}$.

Piezoelectric materials are anisotropic in nature. The directions that are orthogonal to the poling axis are related mechanically by Poisson's ratio, in a similar manner to purely mechanical deformations. Applied deformations in the directions orthogonal to the poling axis will cause an electric field to be created in the poling axis due to the mechanical coupling.

While not truly linear, and containing some hysteresis, the piezoelectric effect can be modeled as such, with little error. The piezoelectric electroelastic relations can be described with the following equation [Ref. 8], shown in stiffness method form:

$$\begin{Bmatrix} T \\ \vec{D} \end{Bmatrix} = \begin{bmatrix} [c] & [e] \\ [e]^T & -[\varepsilon^{\epsilon}] \end{bmatrix} \begin{Bmatrix} S \\ -\vec{E} \end{Bmatrix} \tag{2.1}$$

The stress and strain vectors represent the three axial and three shear stresses and strains. Equation (2.1) can also be represented in an applied force method, given by [Ref. 11]:

$$\begin{Bmatrix} S \\ \vec{D} \end{Bmatrix} = \begin{bmatrix} [s] & [d] \\ [d]^T & [\varepsilon^{\sigma}] \end{bmatrix} \begin{Bmatrix} T \\ \vec{E} \end{Bmatrix} \tag{2.2}$$

These equations, when the [d] and [e] matrices are zero, reduce to Hooke's Law and the dielectric equation [Ref. 9, p. 4]. The relationship between the [d] and [e] matrix is given by the following:

$$[e] = [c]^{\epsilon}[d] \tag{2.3}$$

The dimensions of the [c] matrix are six by six. It contains all of the linear elastic and flexural terms, and is symmetric. The [e] and [d] matrices are six by three in

dimension and contain the piezoelectric coupling terms. Finally, the [ε] matrix is a diagonal matrix with the dielectric permitivities. The individual constants in the matrices are usually indexed by their coordinate axes (1, 2 and 3); the 3-axis is the poling axis.

3. Application and Operating Characteristics

Piezoelectric sensors and actuators are attractive for use for a number of reasons. As a strain gage, a piezoelectric material can be as high as one million times more sensitive than a traditional metal-foil strain gage. They also possess low noise, and being a ceramic material, low to moderate temperature sensitivity. Their bandwidth and response are compatible with their use as actuators. The power required by a piezoelectric actuator is low, but due to the high voltages required, requires special attention when placed in an orbiting spacecraft.

Basic schematics for a piezoelectric sensor and actuator are presented in Figures 3 and 4. The sensors and actuators can be mounted such that the sensing direction and desired applied force can be in the direction of the poling axis, or orthogonal to it. In the following figures, the poling axis and the applied or sensed electric field are shown, as are the displacements in the piezoelectric crystal.

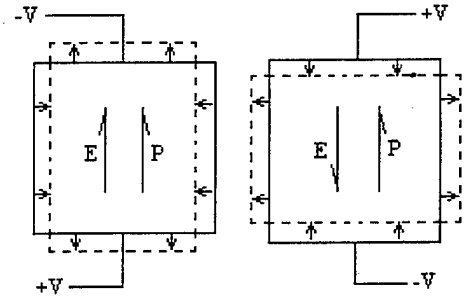


Figure 3. Piezoelectric Actuator Configuration

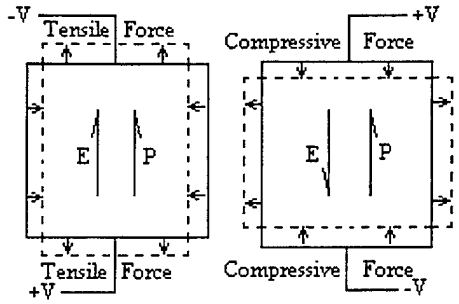


Figure 4. Piezoelectric Sensor Configuration

Piezoelectric materials are limited in that the electric field that can be applied must be less than 2kV/mm [Ref. 12]. To functionally eliminate this problem, several layers of piezoelectric material are stacked in the direction of the poling axis, with interleaving foil electrodes to provide a greater cumulative strain and thereby greater force to the structure. The polarity of the piezoelectric wafers is inverted at each step to simplify the electrode placement and maintain uniform global effect. This configuration for a stacked linear actuator is shown in Figure 5.

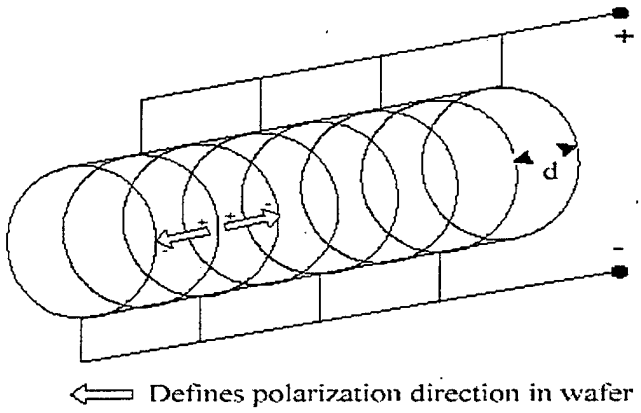


Figure 5. Stacked Piezoelectric Actuator [From Ref. 13]

The active piezoelectric element is mounted in the structure such that the poling axis is directly in line with the element that it is replacing. This mounting will allow the

maximum piezoelectric effect to be felt by the structure, and minimize damage to the brittle ceramic wafers within.

The piezoelectric active control element that was selected was the Piezoelectric Translation Model P-848-30, built by Polytek Physik Instrumente of Hamburg, Germany. The P-843.30 has a maximum operating voltage of 100V. It is cylindrical in shape, with a 14-mm diameter and 73-mm length. The P-843.30 can be seen in Figure 6. Other characteristics are contained within Appendix A.

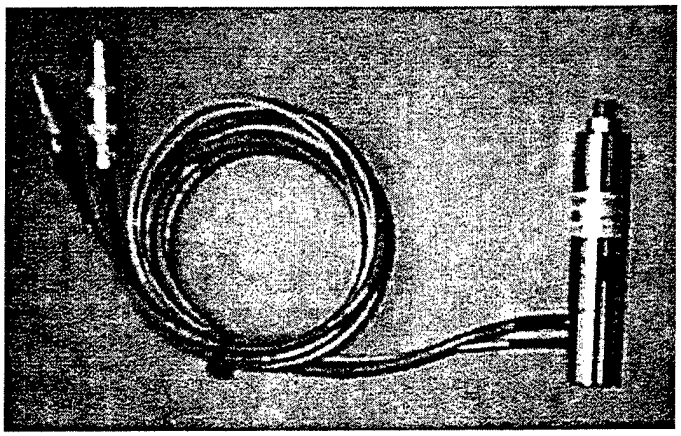


Figure 6. Physik Instrumente P-843.30

The P-843.30 has two connectors, one for the piezoceramic wafers, the other for an integrated metal foil strain gage that can be also used for controls applications. The strain gage was not used for these experiments. A Fitting was added to make the connectors compatible with a BNC connection.

The P-843.30 has a rated open loop travel characteristic of 45 μm/100V +/- 20-percent. An experiment was performed to verify this amount for the actuator prior to installation [Ref. 14, pp. 25-28]. To perform this test, the actuator was mounted to a right-angle stand and a voltage applied. The displacement was measured with a Kaman Eddy Current Sensor. The data recorded was within the manufacturer's stated tolerance

and displays the hysteresis discussed earlier. The results from this measurement are presented in Figure 7. Piezo Model P-843.30 Expansion Characteristics [From Ref. 14].

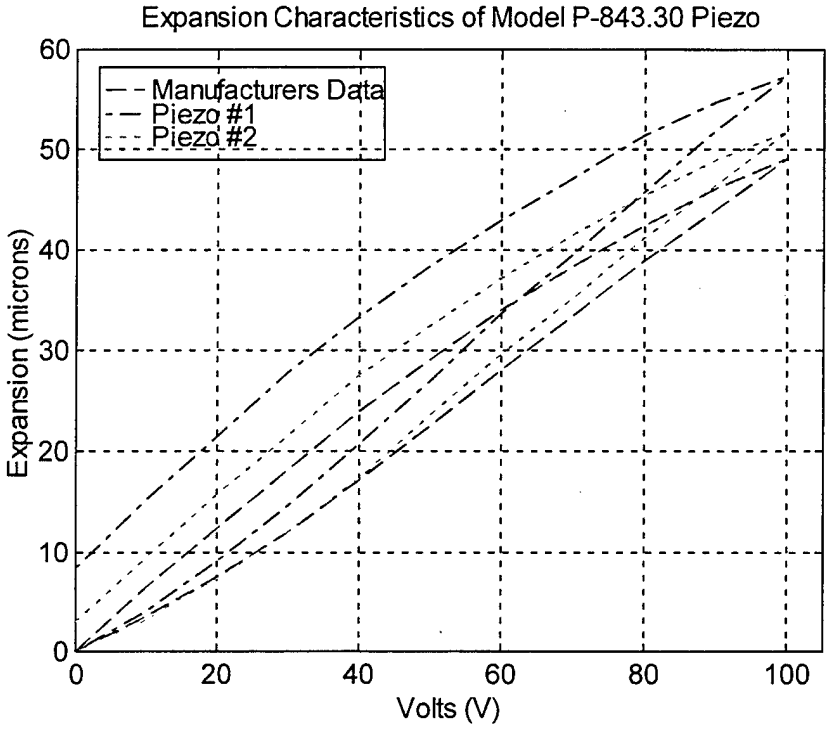


Figure 7. Piezo Model P-843.30 Expansion Characteristics [From Ref. 14]

As can be seen from the previous Figure, there are two piezoelectric actuators. One of the actuators was damaged irreparably during installation; the serial number on the installed actuator can not be read. It was assumed that the average value would suffice for further calculations, at 50 μm/100V.

4. Locating The Active Element

A finite element model of the NPS space truss was constructed in MATLAB to verify the active strut location. It was created using the three-dimensional frame elements found in Kwon and Bang [Ref. 15, p. 264]. These elements are subjected to axial, bending and torsional loads. For a complex shape, like an I-beam, the elements can be

complex. In the NPS space truss, the elements are all tubular and axially symmetric, and some simplifications were used.

The equation of motion for a simple n-degree of freedom undamped system is given by the following equation, shown in matrix tensor notation:

$$[M]\{\ddot{x}\}+[K]\{x\}=\{F\}. \tag{2.4}$$

The stiffness matrix, [K], for a three-dimensional tubular frame element is given by

$$[K^e]=\begin{bmatrix} K_{11}^e & K_{12}^e \\ K_{21}^e & K_{22}^e \end{bmatrix}, \tag{2.5}$$

where

$$[K_{11}^e]=\begin{bmatrix} a_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & b_1 & 0 & 0 & 0 & b_2 \\ 0 & 0 & c_1 & 0 & -c_2 & 0 \\ 0 & 0 & 0 & a_2 & 0 & 0 \\ 0 & 0 & -c_2 & 0 & 2c_3 & 0 \\ 0 & b_2 & 0 & 0 & 0 & 2b_3 \end{bmatrix},$$

$$[K_{21}^e]^T=[K_{21}^e]=\begin{bmatrix} -a_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -b_1 & 0 & 0 & 0 & b_2 \\ 0 & 0 & -c_1 & 0 & -c_2 & 0 \\ 0 & 0 & 0 & -a_2 & 0 & 0 \\ 0 & 0 & c_2 & 0 & c_3 & 0 \\ 0 & -b_2 & 0 & 0 & 0 & b_3 \end{bmatrix},$$

$$[K_{22}^e]=\begin{bmatrix} a_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & b_1 & 0 & 0 & 0 & -b_2 \\ 0 & 0 & c_1 & 0 & c_2 & 0 \\ 0 & 0 & 0 & a_2 & 0 & 0 \\ 0 & 0 & c_2 & 0 & 2c_3 & 0 \\ 0 & -b_2 & 0 & 0 & 0 & 2b_3 \end{bmatrix}. \tag{2.6}$$

The constants in the stiffness sub-matrices reflect the different effects of compression, torsion and bending. The 'a' terms reflect the axial stiffnesses both compressive and torsional. The 'b' and 'c' terms are used for the bending properties of the frame elements. The three 'b' and 'c' terms reflect the superposed sum of deflections from two displaced nodes with zero slope, one displaced node with zero slope and the other with slope but no deflection, and zero deflection but with slope on both ends. These terms are all described below:

$$a_1 = \frac{EA}{L}, \quad a_2 = \frac{GJ}{L},$$
$$b_1 = c_1 = \frac{12EI}{L^3}, \quad b_2 = c_2 = \frac{6EI}{L^2}, \quad b_3 = c_3 = \frac{2EI}{L}. \qquad (2.7)$$

For each element, there are twelve degrees of freedom: six that relate to the three-axis displacements at the two nodes, and six that describe the rotation of the nodes in the three axes. The shape functions for the axial displacement and rotation are linear, while the off-axial displacements and rotations are cubic, Hermitian polynomials, which are independent from each other.

As each elemental stiffness matrix is formed, it is in a local coordinate frame, with the x-axis aligned with the long axis of the element. The y- and z-directions are orthogonal to this in a user-defined direction. The other axes are important when the beam is not symmetrical about the axial direction. To rotate the coordinate frame from the local to the global requires the use of a direction cosine matrix (DCM). The DCM consists of the dot products from one coordinate frame to another, in this case the (u, v, and w) local frame and the (i, j, and k) global frame. This rotation can be seen in matrix tensor notation below

$$\left[K^e\right]^g = \left[DCM\right]^{gl}\left[K^e\right]^l\left[DCM\right]^{l\,g} \tag{2.8}$$

For each node's displacements and rotations, there must be a DCM. The overall DCM then becomes

$$\left[DCM\right]^{gl} = \begin{bmatrix} \left[dcm\right]^{gl} & & & \\ & \left[dcm\right]^{gl} & & \\ & & \left[dcm\right]^{gl} & \\ & & & \left[dcm\right]^{gl} \end{bmatrix}, \tag{2.9}$$

where each individual direction cosine matrix (dcm) is given by

$$\left[dcm\right]^{gl} = \begin{bmatrix} \hat{i}\cdot\hat{u}^g & \hat{i}\cdot\hat{v}^g & \hat{i}\cdot\hat{w}^g \\ \hat{j}\cdot\hat{u}^g & \hat{j}\cdot\hat{v}^g & \hat{j}\cdot\hat{w}^g \\ \hat{k}\cdot\hat{u}^g & \hat{k}\cdot\hat{v}^g & \hat{k}\cdot\hat{w}^g \end{bmatrix}. \tag{2.10}$$

The local coordinate system is defined as having the x-axis aligned with the long axis of the element, with the y- and z- directions orthogonal to it. When writing an algorithm to automate this for a computer program, the u, v and w unit vectors were defined in terms of the global frame. This allowed the cosines (dot products in the [dcm] above) to be easily calculated. In the global frame, the axial direction was defined as

$$\hat{u}^g = \frac{\Delta x}{L}\hat{i} + \frac{\Delta y}{L}\hat{j} + \frac{\Delta z}{L}\hat{k}. \tag{2.11}$$

Since the beams were symmetric, the direction of the y- and z-local directions was arbitrary. This allowed the use of an off-axis vector, which was used in forming the other orthogonal axes, to be chosen by a simple rotation of the x-local direction vector by 90-degrees along the z-global axis. This was useful except when the x-local axis was already.

aligned with the z-global direction, in which case the x-local axis was rotated in the y-global direction. The off-axis vector was defined by

$$\hat{OA} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \hat{u}^g, \text{ or } \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \hat{u}^g. \tag{2.12}$$

The v-direction was derived by obtaining the cross product of the u-vector with the off-axis vector. Similarly, the w-vector was obtained by completing the right-hand rule involving the u- and v-direction vectors. These relations are

$$\hat{v}^g = \frac{\hat{u}^g \times \hat{OA}}{\left|\hat{u}^g \times \hat{OA}\right|}; \quad \hat{w}^g = \hat{u}^g \times \hat{v}^g. \tag{2.13}$$

With the element DCM obtained, the local stiffness matrix may be rotated into the global frame for integration into the global stiffness matrix by using the tensor equation (Equation (2.8)).

The next requirement for structural dynamic analysis is the mass matrix, [M]. There are many approximations available to a modeler for a mass matrix; some of them are lumped, consistent and compound mass matrices. For the NPS Space Truss FEM, all three were attempted.

The lumped mass matrix consists of taking the total mass of the element and assigning half of the mass to each node in each translation degree of freedom. The inertia terms are neglected for a lumped mass matrix analysis. The single-axis form of the lumped mass matrix is given by:

$$\left[M^e\right]^l = \frac{\rho AL}{2}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.14}$$

In a multiple degree of freedom system, the matrix is expanded so that each translational degree of freedom is given one half the mass of the member. The total mass in all the purely translational terms is equal to the total mass of the element, multiplied by the number of translational degrees of freedom per node. This is true for all mass matrices, regardless of construction method.

Integrating the shape functions for each degree of freedom, multiplied by the mass, over the length of the element forms the consistent mass matrix. The matrix tensor form of the consistent mass matrix is given by

$$\left[M^e\right]^l = \int_0^L \rho A\left[H\right]^T\left[H\right]dx, \tag{2.15}$$

where [H] is the matrix containing the shape functions for the element, applied for the six degrees of freedom. The shape function matrix is of size one by twelve for a two-node, six-degree of freedom element. For the linear axial shape functions and Hermitian off-axis shape functions, the [H] matrix is in the form

$$\left[H\right] = \left[H_{1L} \quad H_{1C} \quad H_{1C} \quad H_{1L} \quad H_{2C} \quad H_{2C} \quad H_{2L} \quad H_{3C} \quad H_{3C} \quad H_{2L} \quad H_{4C} \quad H_{4C}\right] \tag{2.16}$$

where the linear and cubic shape functions are those found in Ref. 15. The independence of the axial displacement, axial rotation and off-axis displacement/rotations is apparent from the reuse of shape functions in equation (2.16).

It was found through trial and error that including the small rotational inertia cross-terms in the consistent mass matrix led to floating point errors during the MATLAB model run during eigenvalue solving. Therefore, it was decided to neglect these terms from further analyses. The form of the consistent mass matrix used in the NPS space truss model is given by:

$$\left[M^e\right]^l = \frac{\rho AL}{210}\begin{bmatrix} 70 & 0 & 0 & 0 & 0 & 0 & 35 & 0 & 0 & 0 & 0 & 0 \\ 0 & 78 & 0 & 0 & 0 & 0 & 0 & 27 & 0 & 0 & 0 & 0 \\ 0 & 0 & 78 & 0 & 0 & 0 & 0 & 0 & 27 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 35 & 0 & 0 & 0 & 0 & 0 & 70 & 0 & 0 & 0 & 0 & 0 \\ 0 & 27 & 0 & 0 & 0 & 0 & 0 & 78 & 0 & 0 & 0 & 0 \\ 0 & 0 & 27 & 0 & 0 & 0 & 0 & 0 & 78 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{2.17}$$

The differences found in the axial (diagonal) and off axial (off-diagonal) terms are due to the fact that the axial shape functions were linear, and the other directional shape functions were cubic. For matrix inversion, the zero rows and columns in the mass matrix were removed, and subsequently replaced to restore the degrees of freedom.

Some computer analysis programs, such as MSC/NASTRAN contain an option for a different kind of mass matrix, called a compound mass matrix [Ref. 16]. Taking the average of the lumped and consistent translational terms forms the compound mass matrix. The form of the compound matrix for linear elements is given by

$$\left[M^e\right]^l = \frac{\rho AL}{12}\begin{bmatrix} 5 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{2.18}$$

For cubic shape functions, the form of the compound matrix is different. As an approximation, and to ease computational burden, the compound mass matrix shown above was finally used for the NPS space truss analytical model. Finally, the additional mass from the node balls and fittings was added as concentrated mass, located at the nodes.

The damping effects in the NPS truss were neglected for the MATLAB FEM, as they were not expected to contribute heavily either to the natural frequency or the best location for the control strut. In previous experiments, the damping ratios were found to be less than ten-percent, which would lead to less than one-percent error in the natural frequency. This relationship is shown by the following equation:

$$\omega_d = \omega_n \sqrt{1-\zeta^2} = \omega_n \sqrt{1-0.1^2} = \omega_n \sqrt{0.99} \cong \omega_n . \tag{2.19}$$

For the finite element modeling, the MATLAB m-file "ex895.m" [Ref. 15, p. 276] was extensively modified to form the backbone of the model creation as "Final_root_nps.m." This program is included as Appendix B. In order to automate the calculation of the elemental mass and stiffness matrices, 'feframe2.m' [Ref. 15, p. 278] was modified to provide a three-dimensional element, convert it to the global coordinate frame and provide the mass matrix for each element. This modification is found as 'feframe3.m,' included as Appendix C.

When the root program was run, the eigenvalues of the system were determined. The first ten modes are shown in Table 1, NPS Space Truss Modal Frequencies. For comparison, the natural frequencies for the actual truss, as determined from a modal

testing experiment [Ref. 14] are presented alongside the frequencies obtained in the analytical model.

Mode #	Analytical	Actual	Mode #	Analytical	Actual
1	14.13	14.64	6	72.24	74.54
2	15.44	16.26	7	79.71	80.66
3	28.72	30.41	8	97.41	101.01
4	32.04	33.97	9	120.21	126.23
5	60.23	62.93	10	129.68	135.97

Table 1. NPS Space Truss Modal Frequencies

As can be seen from the table above, there is a small difference between the actual and calculated natural frequencies. There are several possibilities as to where the difference is generated. The modeling of the truss as a series of single elements between the nodes is approximate, and does not include any stiffness for the fittings that join the elements to the nodes. The lumping of mass at the nodes also affected the results. In later, more detailed models, additional stiffness elements could be added for the fitting between the element and the nodes with more model nodes at these fitting locations.

In any true structure, there are infinite degrees of freedom, but this is analytically impossible to model, even for the largest computer. For this model, the results obtained are very close. This is likely due to the high number of degrees of freedom of the system (312), which enables a better representation of the early modes.

The first four mode shapes for the NPS Space Truss were obtained by taking the eigenvector displacements and adding them to their respective nodal coordinates. This will then provide a figure that shows the position of each node in the given mode shape. Each mode shape was created using the code contained in Appendix D, "NPS_modes.m." These figures are included in Appendix E. Modes one and two represent a side-to-side and "see-saw" motion of the truss respectively. Modes three and four represent an "arms

waving" description, sideways and up and down, respectively.

In order to determine the best location for an active control strut, the following approach was used. The structure was analyzed for the first five modes to determine which element had the maximum strain energy for that mode. To target the first mode, the element with the maximum strain energy should be replaced with an active member.

When a structure is vibrating in any of its various modes, each element has some strain energy, caused by displacement along the element length. This energy is at a maximum when the structure is at its largest amplitude. The elemental strain energy is given by the following equation:

$$U^{e}_{strain} = \frac{1}{2} E \epsilon^2 = \frac{1}{2} \frac{EA}{L} \delta^2, \tag{2.20}$$

where

$$\delta^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 \tag{2.21}$$

The change in element length was determined by using the nodal coordinates, when displaced, from the mode shapes determined above. Since eigenvectors are amplitudes for sinusoidal functions, the displacements can be negative as well as positive. This means that the geometrically symmetric elements are also as effective as the truss vibrates in the other direction through a cycle. The numerical results for the relative strain energy determination were generated using code contained in Appendix F.

Due to the diagonal elements in the truss, there is an odd symmetry to the locations of the maximum strain energy. The highest elements occur near the center of the truss in all modes except the third (torsional) mode. With this information the location for the active control strut may be selected. Element 101 is the location with the maximum strain energy for mode one. A second piezoelectric control strut can be placed

on the opposing side of element 107. This is possible due to the symmetry described above and the desire to not have both actuators acting through the same node ball.

5. Force Transducer

A large issue for any active structure's design is the location of the sensors and the actuators [Ref. 15, pp. 460-461]. The vibrational motion of a structure involves phasing difference that can vary for all frequencies that a structure will experience. For this reason, a co-located sensor is attractive, as it will read the same phase as the applied force. In a noiseless system, a co-located sensor can mathematically provide unconditional stability, given proper control law. It also provides for analytical simplifications due to the application at the same point as the sensed output.

The force transducer that was selected was the PCB Piezotronics Model 208B02 General Purpose ICP Force Sensor. This sensor is a linear, piezoelectric device, with quartz as the piezoelectric material. It is threaded on either end so that it can be placed in the structure to experience an axial load. It is capable of detecting a force of up to 1000-lb compressive under static conditions and 100-lb under dynamic conditions [Ref. 17]. The device is 15.75-mm long and approximately 17-mm in diameter. The model 208B02's operating characteristics are included in Appendix A.

6. Installation of the Active Element [Ref. 14]

Once the active strut components were gathered and tested, they were assembled into a single active truss element. Machined steel rods were created to provide an interface between the active element and the truss. Given the susceptibility to moment

and torsional loads, a PI flexible tip was inserted between the sensor and actuator. The assembled active element is shown in Figure 8.

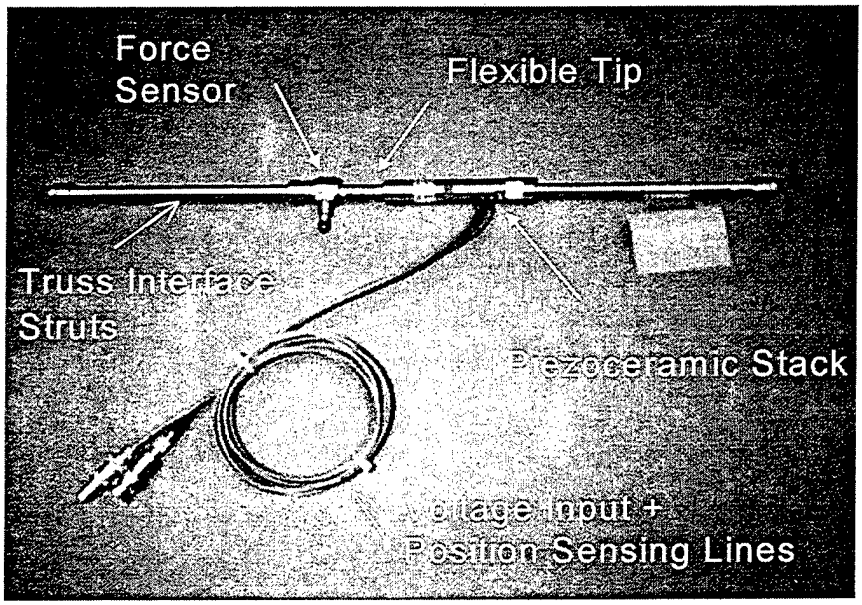


Figure 8. Assembled Active Control Element [From Ref. 14]

To utilize the full range of the piezoelectric device, a preload was required. The options were to mechanically preload the device, at the cost of loading the structure, or to determine a means for an electrical preload. Mechanical preloading was used and was achieved by inserting shims into the element to node ball interface until the desired preload was achieved. Full details on the installation of the active control element can be found in Reference 14.

During installation of a second active control strut, excess torsional load was applied, resulting in failure of the second strut. In the future, care must be take to ensure that torsional loads are balanced during assembly. The actuator is machined to allow the use of an open-ended wrench for this purpose.

C. LINEAR PROOF MASS ACTUATOR (LPACT) [Ref. 14]

In order to provide a disturbance force, a Linear Proof Mass Actuator (LPACT) was installed in the NPS space truss. The LPACT is a model CML-030-020-1 manufactured by Planning Systems, Incorporated, from Melbourne, FL. The LPACT is powered by a separate amplifier and controller assembly that has embedded feedback electronics. These feedback loops, while not used for these experiments, could also be used for active damping control. The LPACT also has mounted accelerometers for use in monitoring or driving the feedback loops. Details on the LPACT are contained within Appendix A. The LPACT is shown in Figure 9.

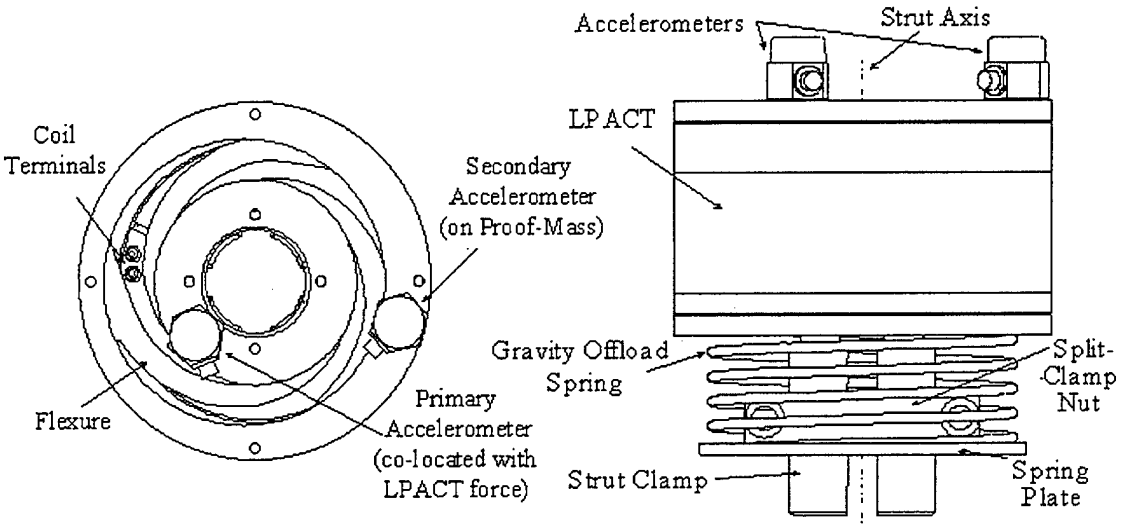


Figure 9. LPACT [From Ref. 18]

The LPACT was mounted in a similar manner as the active strut. The device was balanced to locate its center of mass, to ensure that symmetry would be maintained in the truss and to prevent a torsional static load down the length of the truss. Aluminum interface struts were constructed and attached to the LPACT for mounting. It was placed in the truss in the outermost diagonal, to simulate a payload that is vibrating at critical

frequencies. After mounting, the Gravity Offload Spring was used to return the LPACT assembly to the center position, to offset the weight of the mass at a 45-degree angle. A picture of the installed LPACT is provided as Figure 10.

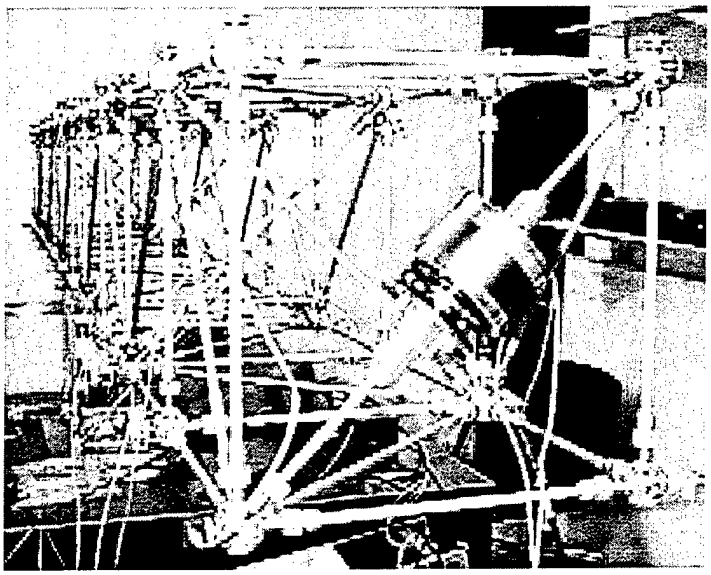


Figure 10. LPACT Mounted on NPS Space Truss [From Ref. 14]

The LPACT, when the internal feedback loops are disabled, is powered by an external signal, from a signal generator, or other source. As the LPACT is a real device, it has its own frequency characteristics. A description of the output force and phase are shown in Figure 11. It is of note that the LPACT has its natural frequency at about 8.5-Hz, and a tapering off of the amplitude from that point. For the experiments to follow, the range of 10 to 20-Hz is of particular interest. The mass of the LPACT does contribute to a shift in natural frequencies of the truss. This is in addition to the inclusion of a new natural frequency in the 8-10 Hz range that is the product of the LPACT spring.

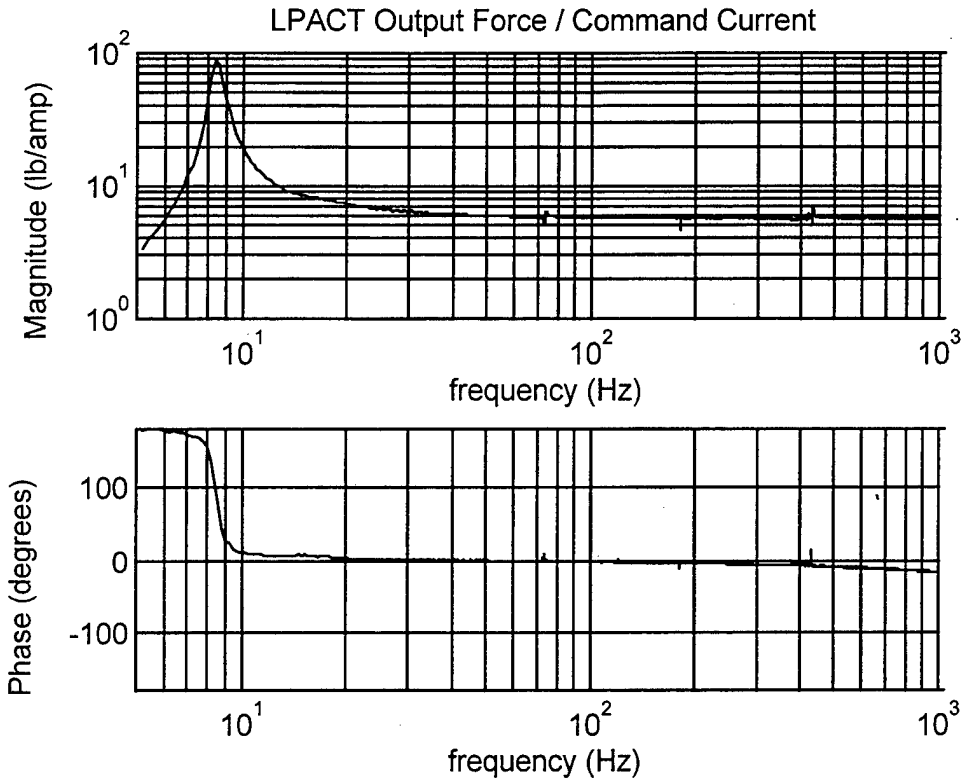


Figure 11. LPACT Transfer Function (feedback loops off) [From Ref. 18]

D. LASER DIODE ASSEMBLY [Ref 14]

To allow a qualitative measurement of the truss' vibration when under the influence of the LPACT, a laser diode was mounted to the end of node 52, such that it would project a beam onto the wall. The mounting consisted of a thin aluminum rod, with a mass at the end, housing the laser. The laser diode selected was a 1-mW, 635-nm Model PLC6351FW from Lasermate Corporation of Walnut, CA. It is powered by a Hewlett Packard E3615A DC Power Supply, operating at 2-4 Volts nominal. A picture of the laser diode assembly is included as Figure 12. Full details on the laser diode can be found in Reference 14.

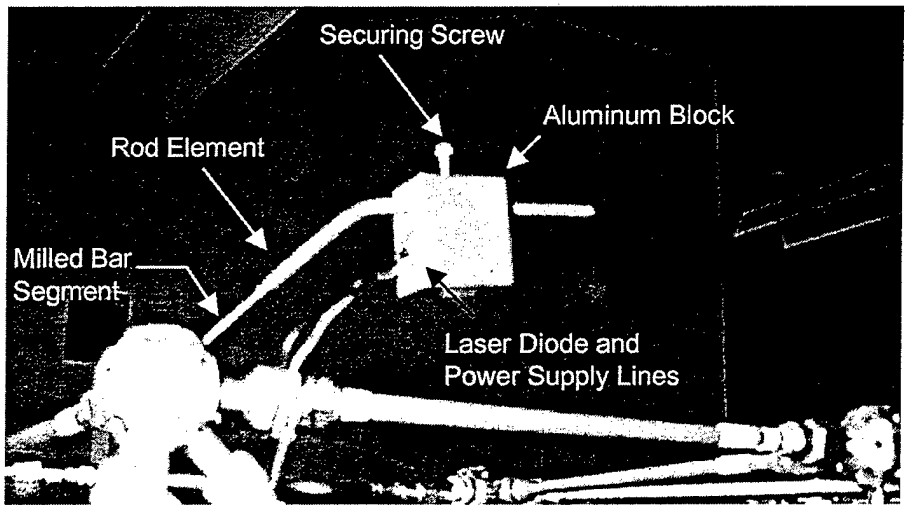


Figure 12. Laser Diode Assembly

E. PREVIOUS EXPERIMENTS

Two previous theses have been completed on the NPS space truss: One by CAPT Brent K. Andberg, USMC, and the other jointly by LT Scott Johnson and LT John Vlattas, USN.

1. Andberg [Ref. 19]

In his thesis, CAPT Andberg developed an FEM of the NPS space truss using an NRL code entitled NRLFEMI. He then performed modal testing on the truss to confirm the model. The experimental data was lacking in that it failed to observe the first mode. Finally, CAPT Andberg performed a technology demonstration of the use of Fiber-optic Bragg Gratings (FBG), used in this example to detect the motion of a simple cantilever beam. In the future, FBGs will be installed on the NPS space truss for shape determination.

2. Vlattas and Johnson [Ref. 14]

This Master's Thesis focussed on two areas, the re-performance of modal testing with the HP 35665A two-channel spectrum analyzer and integration and testing of an active control device. They recommended that modal testing be again done, to overcome some of the limitations of the dSPACE system that was installed at the time. As will be discussed in the next chapter, a new dSPACE system was installed on the NPS space truss, and would be available for this purpose, if desired.

The active controls integration produced good results, with a maximum reported reduction of 14.817 dB at 16.85 Hz. They reported using a disturbance amplitude of 100 mV for the LPACT source. It was determine early on through the current course of research that this did not even provide a sensor signal sufficient to overcome system noise. Therefore, the 14.817-dB reduction in amplitude is held in question. One of the experiments performed later will be to validate these results with the new dSPACE system.

III. CONTROLS SYSTEM INTEGRATION

A. OVERVIEW

The active vibration controls system was created with the use of a dSPACE DS 1103 PPC Controller running the control law implemented by Vlattas and Johnson [Ref. 14]. The DS 1103 has a Motorola PowerPC 604e microprocessor as its central processing unit (CPU), and resides in a triple-wide ISA slot in a host PC [Ref. 2, p. 12]. The dSPACE system described in References 14 and 19 was originally shared between several systems; the DS1103 system was ordered and installed for the sole use of the NPS space truss.

The DS 1103 board has connectors for an external input / output (I/O) box that contains BNC fittings and standard computer cable connections for analog to digital (ADC), digital to analog (DAC) and support for other cabling formats (e.g. RS-232). Control of the dSPACE CPU and access to its memory (128MB) is done with the use of ControlDesk, the dSPACE main program, or by using MATLAB / SIMULINK programs or C-code. These programming components are discussed in grater detail below.

B. INTEGRAL PLUS DOUBLE INTEGRAL CONTROLLER

The selection of the integral plus double integral force feedback (IDIFF) controller was due to its inherent stability. A simplified block diagram of an integral controller acting on a unity system is shown in Figure 13. A plot of the magnitude versus frequency for this system is presented in Figure 14.

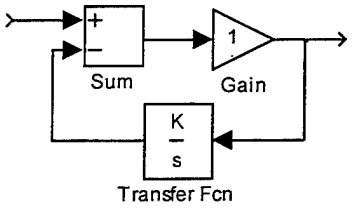


Figure 13. Integral Feedback Controller (Simplified) [After Ref. 5, p. 8]

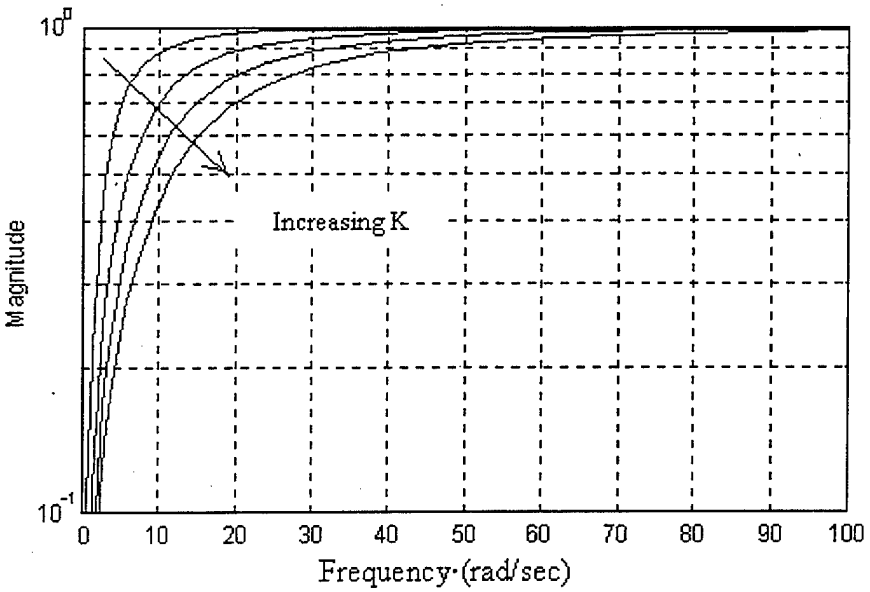


Figure 14. Effect of Integral Gain on Closed Loop Transfer Function [After Ref. 5, p. 8]

In theory, the controller gain would be increasingly able to control the system output. However, the authority of the actuator (how much force it can impart to the structure) and the noise in the system lead to a point where the maximum gain is achieved [Ref. 5, p. 8]. Further, the addition of digital filters to the design to attenuate the DC bias will add additional factor that also invalidate the unbounded controllability inherent in the integral controller. The use of the second integrator can be used to finely adjust the phase response of the controller, and allow a slightly better response than the single integral alone. This result will be discussed in more detail later in the thesis.

C. SIMULINK / RTI

1. Simulink Controller

The first step in creating the controller was to assemble a Simulink model. The Simulink model contains the processing connections necessary, implemented with block diagrams, to create the control law. The original controller, found in Reference 14, was recovered and updated for the current version of MATLAB and Simulink.

The original controller contained two band-pass filters designed to isolate a desired frequency (in their case, 16.75 Hz). It was decided to test the controller over a larger range of frequencies, therefore the band-pass filters were at first removed. It was found through several test runs that the input and output, which will be fully described in the next subsection, contained a DC bias that needed to be removed, lest the integrators give an unbounded output. Therefore, it was decided to implement a high-pass filter in the signal path. This need is consistent with the results observed by the researchers at NRL, who used a second-order Butterworth filter to remove the DC bias [Ref. 5, p. 6]. Higher order filters were tested to determine their utility. It was discovered that the higher-order did not lead to an appreciable increase in filtering, but did decrease the instantaneous response time. A third-order filter was used for the NPS space truss.

The Simulink model also contains a provision to prevent supplying an over voltage to the piezoelectric strut. A saturation limiter with DC bias is included as part of the control signal, before it reaches the output block.

The dSPACE system relies upon the Simulink model for the selection of its sampling frequency. It was determined that frequencies above 250 Hz (as reported in

Chapter V, section B.5) were not consequential. Therefore, a sampling frequency of 500Hz was utilized, yielding a sample time of 0.002 seconds. When the Simulink model is complete, the command "Tools > RTW Build" will create the real-time program and object file that can be executed by the CPU of the dSPACE system.

2. RTILIB

The interface from Simulink to dSPACE is found in the Real-Time Integration Simulink Library (RTILIB). It is invoked by typing "rtilib" from the MATLAB command window after Simulink is running [Ref. 20, p. 17]. The RTILIB contains all the additional blocks required to provide the interface between the Simulink model, running on the CPU, and the external I/O box. It also provided blocks for hardware interrupts for more complex programming needs. The RTILIB interface is shown in Figure 15.

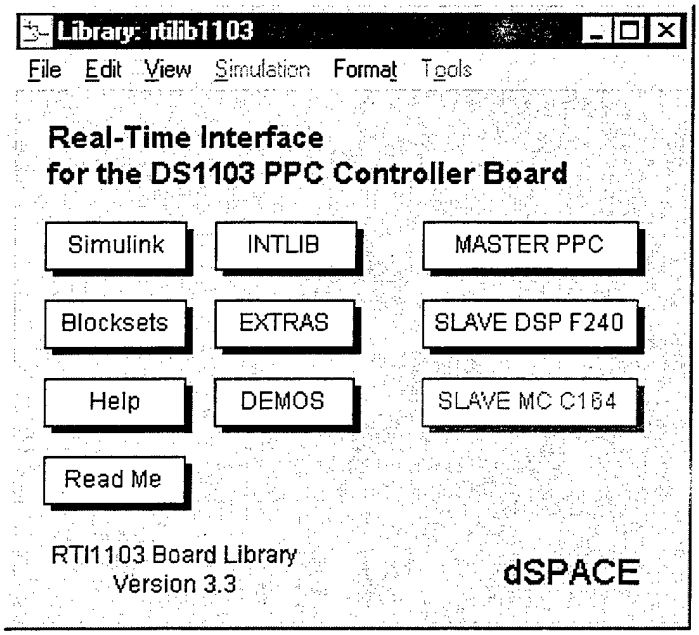


Figure 15. RTILIB Interface [From Ref. 20, p. 18]

To use the blocks in a model, they are selected and then "dragged and dropped" onto the Simulink model for integration. The I/O box connections are found under the "MASTER PPC" icon. There are twenty available inputs and eight outputs available to the user in dSPACE. They are accessible by three blocks, shown in Figure 16.

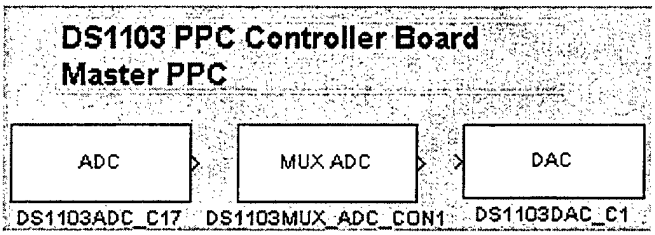


Figure 16. MASTER PPC I/O Connections [After Ref. 20]

The ADC channels available are the ADC and MUX (Multiplexed) ADC, shown above. The MUX ADC is a series of four channels, with four multiplexed inputs each (ADCH 1-16). They can be used in Simulink as individual separated channels, or as a vectored input. The scalar ADC block is a single input, corresponding to the last four input channels (ACDH 17-20). In similar manner, the eight output channels are accessed with the DAC blocks (DACH 1-8) [Ref. 20, p. 24]. The I/O blocks, when connected to the I/O interface box, exhibit a 10:1 gain in value. For example, a 0.5 scalar value in the running model will produce a 5.0-Volt output at the DAC.

Additional gain blocks were added to the model to allow for turning on and off the controller without modifying the gain settings. Also, vectored input blocks were added to allow the capture of the four accelerometers used in the experiments. The final controller block diagram is shown in Figure 17. NPS Space Truss Controller.

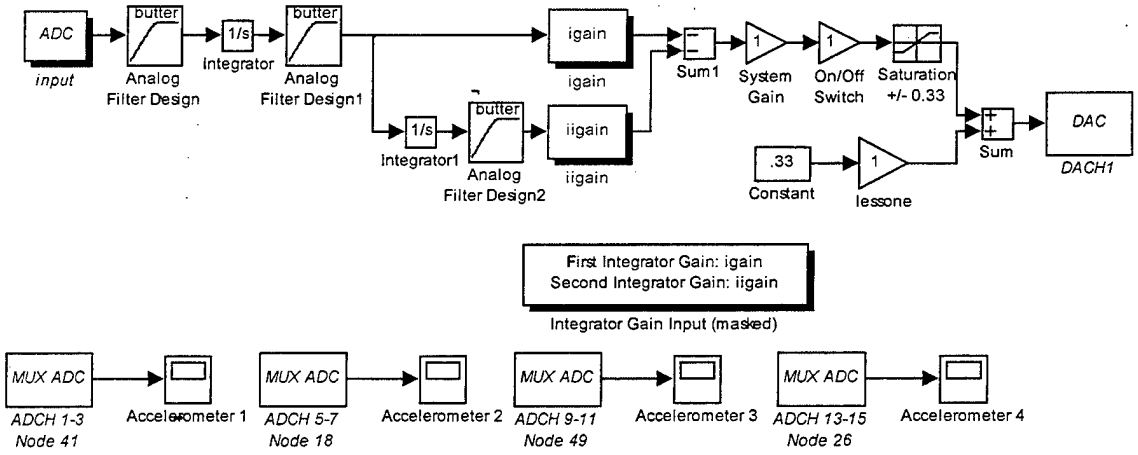


Figure 17. NPS Space Truss Controller

D. dSPACE CONTROL DESK

In previous versions of dSPACE, data input and output was managed with the use of two different software packages: Trace and Cockpit. Trace allowed for real-time detection of the signal and other variables with software capture, Cockpit permitted quasi real-time adjustment of the parameters running on the CPU. In the latest versions of dSPACE, these programs have been melded into a single entity, called ControlDesk, and expanded in scope to include some of the Windows-type features that the majority of computer users have become familiar. In addition to allowing the access and variation of data in the CPU, ControlDesk also allows for grouping relevant files under an "experiment" and running of macros to automate the data taking process using the dSPACE macro language, Python. [Ref. 21, pp. 13-14]

ControlDesk's most useful improvement is the addition of Windows "drag and drop" (D&D) capability. It allows for the selection of variables for real-time display in on-screen instruments, and for entire programs to be downloaded and executed by the CPU [Ref. 21, p. 31]. To establish a working experiment, first a program must be loaded

into the CPU (or the CPU's memory), then an instrument panel may be built that will monitor and control the parameters and signals desired. The ControlDesk interface contains several sub-windows that allow a user to run his/her experiments. These windows are the main window, Navigator, and Tool Window. The Navigator allows file and program manipulation; the Tool window allows the selection and control of the various variables in a model. The main window allows for the controlling of the simulation (i.e. running the CPU) [Ref. 21, pp.43-50].

Instrument panels are built using a pull-down menu interface and using D&D to place the desired instrument on the main window. There are many different types of instruments available, to simplify the visual arrangement of the experiment to the user. Some of the instruments available are sliders, pointers, oscilloscopes, knobs and buttons. Selecting the proper variable is important. The variables can be of many different types, and are grouped according to their block location in the original Simulink model. The relevant types to this thesis are shown in Table 2.

Type	Description:
B:	Block Outputs
S:	Inputs of Signal Sinks
P:	Block Parameters

Table 2. dSPACE Variable Types [After Ref. 22]

The controller layout used for the IDIFF controller on the NPS Space Truss is shown in Figure 18.

The four large plots on the top of the Layout are oscilloscope plots of the four accelerometers, with the three axes measured superimposed on each plot. The plot in the lower right-had portion of the layout is used to track the incoming signal from the force sensor and the signal to be output to the actuator. The lower left-hand portion of the

layout contains controls for the various gains and a read or green light to indicate from a glance the controller status (controlling or uncontrolled).

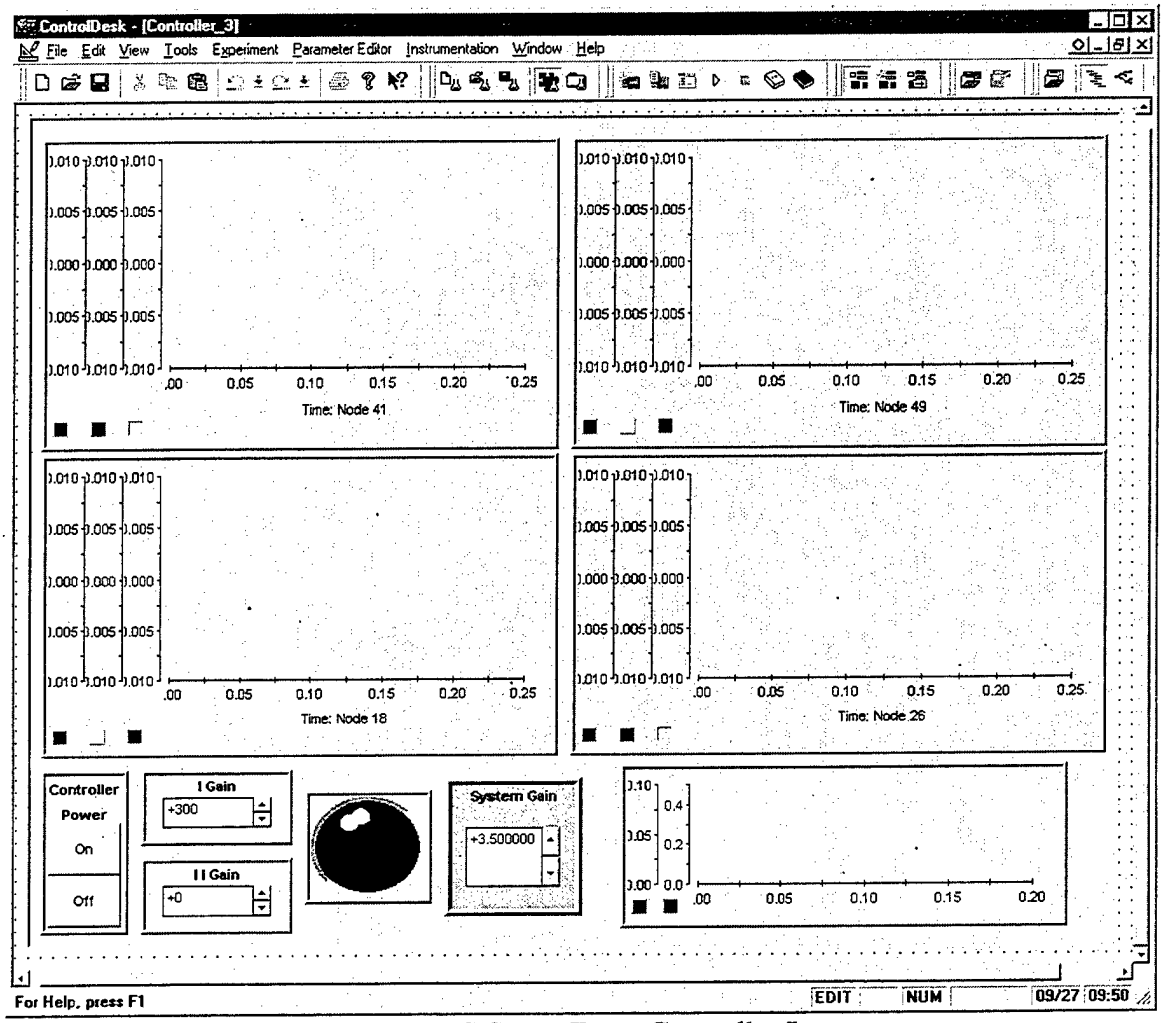


Figure 18. NPS Space Truss Controller Layout

To start and run the controller, the following steps must be followed (all on the NPS space truss computer in the Smart Structures Laboratory):

- Initiate ControlDesk using the icon either on the desktop of the toolbar, or by using the menu Start>Programs>dSPACE>ControlDesk
- Open the Experiment using the command File > Open Experiment. Select C:\Space_Truss\truss99\Space Truss.cdx.

- Load the controller to the CPU by using the  icon on the toolbar. Select the program C:\Space_Truss\truss99\ST_Controller_3.ppc (the number refers to the third iteration of the controller).
- Ensure the Edit Mode is selected from the toolbar (verify the Edit mode button depressed)  . The three buttons are Edit, Test and Animate mode respectively.
- Start the CPU (depress the green triangle on the toolbar).
- Select the Animate mode to enable the display.
- The controller may be turned on and off with the push buttons in the lower left corner of the layout.

E. AMPLIFIERS AND CONDITIONERS

The final segment of the NPS space truss controller consists of the amplifiers and signal conditioners that are external to the PC and the dSPACE system. These components are described below. The signal that is produced by the force transducer is fed into PCB Piezotronics Model 484B signal conditioner, which has a unity gain and a selectable bias (either 6.0 or 11.0 Volts, DC or AC). The four accelerometers used have three-axis capability. Each axis from the four accelerometers is fed into a 12-channel Kistler Piezotron Coupler, Model 5124A. Finally, the control signal is sent to a Trek 50/750 Voltage Amplifier. Details of the equipment are included in Appendix A. NPS Space Truss Characteristics and components.

The Trek amplifier was due for calibration in September 1998. Due to the timeliness involved, it was decided to not perform a full calibration on the Trek amplifier and instead perform a mini-calibration on it to verify its linearity and set points. The Trek was calibrated for a 20-time voltage gain. The gain verification was done by supplying a voltage from 1.0 to 6.0 V to the amplifier and measuring the output voltage. The details of the mini-calibration are presented in Table 3.

Input Voltage	Output Voltage	Gain
0.998 V	20.26 V	20.3
2.006 V	41.2 V	20.5
3.00 V	59.8 V	19.9
4.03 V	81.4 V	20.2
4.99 V	99.1 V	19.9
5.98 V	119.5 V	20.0

Table 3. TREK Voltage Amplifier Mini-Calibration

The deviation from linearity in the data obtained was deemed to be minimal; therefore the mini-calibration was determined to be sufficient to allow use of the Trek Voltage amplifier for the experiments.

IV. CONTROLLED FINITE ELEMENT MODEL OF THE NPS SPACE TRUSS

A. OVERVIEW OF ANSYS

ANSYS is a powerful finite element utility that has many capabilities. The program can involve many different physical properties, from over 100 types of elements. In this manner, the user can model systems that involve, for example, mechanical, electromagnetic, thermal, piezoelectric and electric characteristics. These options available are based on the licensing obtained from ANSYS. ANSYS can couple these effects into one single model, and provide insight into it inner workings. For the work in this thesis ANSYS version 5.5.2, Multiphysics was used.

ANSYS may be run from a graphic user interface (GUI) or purely by text commands. In the GUI, there are several menus and toolbars that are present to allow a user to perform a number of tasks. These menus and toolbars are as follows:

- Utility Menu: Allows for the control and manipulation of variables, files and graphic displays.
- Input Window: An interface for typing in text commends directly while in the GUI.
- Toolbar: Contains standard and user-programmed macro buttons.
- Graphics Window: Shows visually the desired components, plots and perspectives.
- Main Menu: Allows control of the modeling functions of ANSYS, to include the preprocessor, the solution processor and the two postprocessors.

The user is able to modify the position and size of any of these windows and toolbars, therefore they will not shown here. When commands are referred to in the remainder of this thesis, they will be given in both the GUI menu-based commands and the direct text commands. An example of this is:

Main Menu>Preprocessor>Material Props, etc, Typed Command

where the ">" symbol indicates the use of a menu nested inside the previous one. To execute a desired command, there are usually several options available. For simplicity, these GUI commands will be given in a manner that is logical for the creation of the active model. In this thesis, only the commands that affect the model are included. Commands that affect the graphical display are omitted for spatial considerations. The syntax and specifics or the ANSYS commands used are detailed in Reference 25.

The ANSYS preprocessor permits the creation and manipulation of a finite element model. This model can then be acted upon with loads and restraints, and then meshed into a finite element grid. The solver sub-program allows the selection and specification of different solution types, for example static analyses and modal analyses. For this thesis, a modal analysis was performed on the bare truss, and a transient analysis was performed on a controlled model. ANSYS has two different post-processors, one that works for single cases (or single steps of a transient analysis), and another that allows fusion of the entire transient picture. These post-processors are called the POST1 and POST26 processors, respectively.

When using the GUI for modeling and analysis, the user can also access a log file that gives the text commands that correspond to the graphic commands used. This can enable the user to recover a portion of lost data, or to easily recreate the model with

different parameters. The user can also access an error file directly from the GUI as an aid in resolving model problems.

B. EXAMPLE MODEL

In order to develop and demonstrate the ability to create an FEM of an actively controlled structure, and then be able to control it, a simplified model of a control strut was used. This simple model consisted of two piezoelectric elements, both of the same size and type, one to be used as a sensor, the other as an actuator. These elements were connected to a rigid base at one tip with BEAM4 elements and subjected to an axial sinusoidal force at the other tip. The example FEM, called "piezo5" is shown in Figure 19.

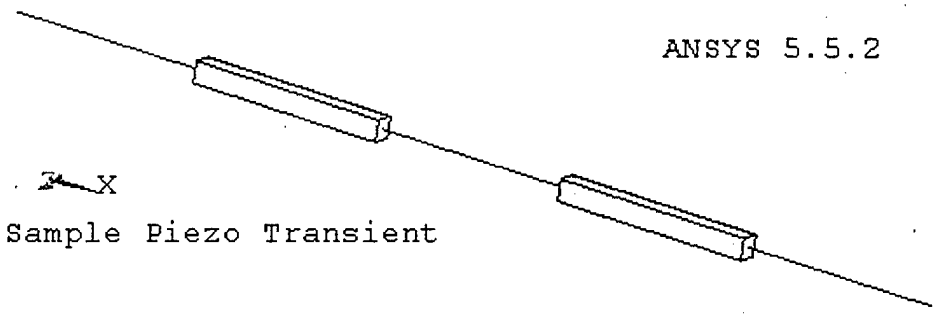


Figure 19. Piezo5 Example Model (Screen Capture)

This model was used only to verify that the algorithm worked properly for an actively controlled device, and was not intended to provide any concrete results. The sections that follow describe in detail how piezo5 was created and modeled in a transient environment, both uncontrolled and controlled.

1. Sensor and Actuator Integration

The ANSYS element that has the required piezoelectric properties is the SOLID5 element. This element allows coupling of many of the physical properties permitted by ANSYS. For the piezo5 demonstration model, two SOLID5 elements were required, one as sensor and the other as the actuator. To simplify the implementation, the same material and geometric properties were used for both elements.

The SOLID5 element has eight nodes; therefore, linear shape functions were used to derive the local elemental matrices for use in the global FEM. Each of the eight nodes has up to six degrees of freedom: three-axis displacement, voltage, temperature and magnetic displacement [Ref. 23, p. 4-41]. It is of note that rotations are not included in the list of degrees of freedom for the SOLID5 element.

The elements were integrated into the model by creating them adjacent to the structural elements that would support them. A rigid region was then created that would couple the degrees of freedom of the piezoelectric element to the end of the BEAM4 element that formed the structure. The command line for a rigid region is:

Main Menu>Preprocessor>Coupling/Ceqn>Rigid Region CERIG

The rigid region was created only in the displacement degrees of freedom, as the SOLID5 element does not have rotations allowed at the nodes. Since the piezoelectric sensor and actuator are designed to detect and apply axial loads only, this was determined to be a sufficient modeling technique. A piezoelectric device is usually only shrouded, not constrained, in the off-axial direction. The rigidization of the ends of the SOLID5 elements will modify its operating characteristics somewhat as the strain induced by fixing the cross-section of the element will reduce the strain applied in the axial direction

somewhat. Follow-on work should attempt to characterize this modeling effect, or determine a resolution.

Piezoelectric effects are, by convention, referred to the poling axis, which is usually the third axis of a Cartesian coordinate system. For the creation of this model, this convention was maintained. ANSYS allows the definition of a local coordinate system that will describe the specific alignment of an element. The local coordinate system can be created by reference to model keypoints, nodes, or the work plane origin [Ref. 22, CS]. The location is irrelevant so long as the alignment is correct. The work plane was relocated and aligned to the axial direction of the active element. The command line to create a local coordinate system aligned with the work plane origin is:

Utility Menu>WorkPlane>Local Coordinate Systems>
Create Local CS>At WP Origin CSWPLA

To create the piezoelectric effect, a voltage difference needs to be applied to the elements. In order to do this a fixed voltage degree of freedom (DOF) was placed along one of the planes perpendicular to the poling axis. To preserve the axial symmetry, and to be consistent with the actual device, the nodes on the other perpendicular plane were constrained to be equal to one another by the command:

Main Menu>Preprocessor>Coupling/Ceqn>Constraint Eqn CE

The final step in creating the model is the application of the specific piezoelectric properties. The SOLID5 element obtains the piezoelectric properties by the use of embedded tables. The piezoelectric table is a six by three table that represents the tensor between the forces and applied voltages, the [e] matrix. The table must be activated before setting the constants, and is referenced row by row, in a sequential manner (the

first row is 1-3, the second 4-6, etc.). Of note is that if any element has a voltage DOF, all or none must have a piezoelectric table. A suggested solution is to have very small piezoelectric constants for the non-piezoelectric elements. [Ref. 23, SOLID5]

The piezoelectric properties used for the Piezo5 model were obtained from a verification model that was included in the ANSYS software package. The specific verification model was VM175, contained in the directory ANSYSROOT\DATA\VERIF. VM175 contains the commands in text format for creating PZT-5A, an effective piezoelectric material. To prevent the model from searching for the magnetic and thermal properties, in an attempt to solve these DOF, the relative permeability and thermal conductivity were set at 1.0. These added constants showed no effect on the results.

2. ANSYS Parametric Design Language (APDL)

It was desired to have a closed program, or a model that could perform an actively controlled data run, requiring no input from the user save to establish initial parameters and start the simulation. Two options for modeling the control law were explored. They are described in the paragraphs below.

The first option involved the use of COMBIN37 Elements operating on model node points to provide a "black box" addition to the model to simulate the controller. The COMBIN37 element has the capability of acting (expanding or contracting) on a derivative, double derivative or single integral of a node point value [Ref. 23, COMBIN37]. The element would be external to the model, fixed at one end, with a constant force applied. The stiffness of the element would be varied according to the integral of the sensor voltage, given by a changing, potentially non-linear parameter. The

changing parameter is described by the term RVMOD; its effect for integrating is given by the following equations:

$$\Delta x = F / k, \tag{4.1}$$

$$CPAR = K \int_0^t \Delta x dt = K \int_0^t A \sin(\omega t) dt = -K \frac{A}{\omega} \cos(\omega t), \tag{4.2}$$

$$RVMOD = RVAL + C_1 |CPAR|^{C_2} + C_3 |CPAR|^{C_4} \tag{4.3}$$

where RVAL is the current real constant value and RVMOD is the new value, when modified by equation (4.3). This leads to the output length of the element, shown here as:

$$\Delta x = F / RVMOD. \tag{4.4}$$

The changing length of the element would supply the input for a LINK11 element imbedded in the structure, serving as the actuator. The advantage for this model would be the speed of implementation, as there would be no external interfaces required by the user, save to modify the gain parameters. Due to the observance of the absolute value terms, this method abandoned as not immediately feasible for sinusoidal sensor input due to the mathematical implementation problem. Later work to allow operation on a bias would remove this problem for a sinusoidal signal.

The second option, and the one that was eventually used, was ANSYS Parametric Design Language (APDL). APDL is a macro language that enables a user to run command streams from an external file. An APDL file is a text file that contains the commands as they would be typed into the ANSYS command interface with user determined variables [Ref. 22, /INPUT, *SET]. APDL was used in conjunction with the transient analysis capability to perform the control law application. ANSYS gives the user the ability to examine each step in a transient analysis; at the time of the step, the

APDL program can read the voltages from a sensor, perform a numerical control law function and apply a voltage to the actuator.

APDL was originally created to allow the variation of model parameters in an optimization design. A capability is present to do automatic optimization, which was not explored at this time. APDL also contains some basic program control flow logic, such as if-then statements and loops. These program control functions were exploited in the creation of a working controller. In order to use an APDL file in a loaded model, the command is as follows:

Utility Menu>File>Read Input From: (select file), /INPUT.

Up to 20 APDL subroutines may run, nested within one another [Ref. 22, /INPUT]. This number is reduced as loops are implemented, as ANSYS a level of nesting for them. This nesting utility allows for more complex programming from within ANSYS to suit the user's desires. In this case, only one input file was required. Local parameters to these nested levels are possible by clearing the new variables before exiting the nested subroutine.

To run the model, a routine was created that would automatically time-step through a transient analysis. A sinusoidal force was applied to the tip. This varying force was reapplied at each step with the use of an array-type variable, indexed to the time step number.

Variables are created and deleted with the use of the *SET command. Variables may be created as scalars, arrays and multi-dimensional tables. The sinusoidal applied force was created with an array that was sized based upon the length of time for the model run. The force was created as an array and as a one-dimensional table. The array

allows accessing by automatic indices. The tables are not pre-indexed, and cannot be accessed by the APDL program without programming the indices. However, for accessing the data in an output file, the tabular format provides simplicity; for this reason, both were used.

3. Control Law Application

The control law was implemented with the use of finite difference equations. In order to approximate the integrating function, a trapezoidal rule was used. The trapezoidal integral approximation is given by the following equation:

$$Y_{t+1} = Y_t + \frac{\Delta t}{2}(X_{t+1} + X_t) \tag{4.5}$$

To perform this algorithm in the APDL macro, storage locations needed to be established for the single and double integrals, one for each.

In simple runs of the model, it was found that the minute positive error from the double precision variables used by ANSYS caused a constant DC bias to be present, which destabilized the model. ANSYS tends to view the real number 0.0 as a random number on the order of 10^{-31}. In order to prevent this occurrence, a simple digital high-pass filter was inserted in the system after the first integral term. This digital filter used a single pole and zero at 1.0 + 0.0i and 0.95 + 0.0i respectively. The transfer function for this filter is given by the following equation:

$$H(z) = \frac{(z-1)}{(z-0.95)}. \tag{4.6}$$

Equation (4.6) is converted into a finite difference equation by the following equation [Ref. 25, p. 97]:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{L} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}} \tag{4.7}$$

Combining the previous two equations gives the finite difference equation to be inserted in the control signal path after the first integrating term, as shown in the following equation:

$$Y(n) = X(n) - X(n-1) + 0.95Y(n-1) \tag{4.8}$$

The frequency response of the digital filter is shown by Figure 20, which was generated with the MATLAB FREQZ command. As can be seen, there is not an appreciable phase error above 5.0- percent of the Nyquist frequency that is inserted by the use of the high-pass digital filter. The magnitude reduction is negligible above 2.0-percent of the sampling frequency. The need for a filter in Piezo5 is consistent with the implementation of the dSPACE digital signal processor described in the previous chapter.

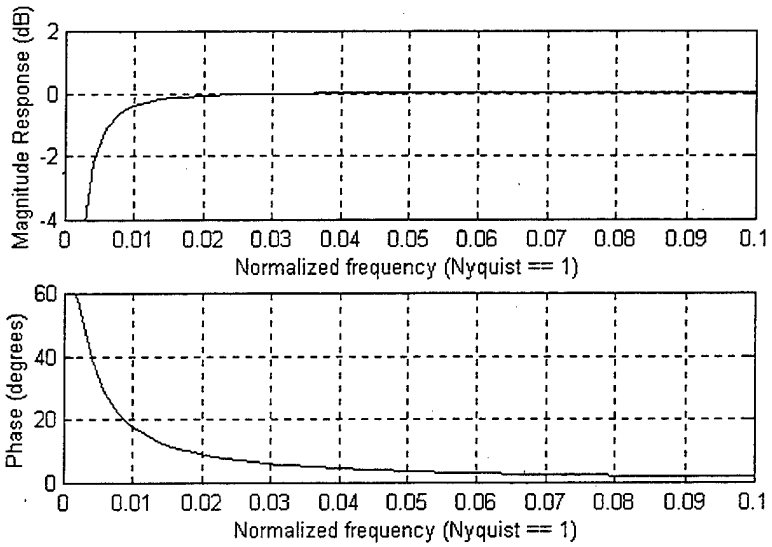


Figure 20. Simple Digital High-Pass Filter Frequency Response

Finally, the control law can be obtained as a combination of the filtered integral term, and a double integral term, derived from the first integral. These terms are multiplied by three gain constants to get the control signal to be applied to the actuator.

The control law is given by:

$$CS = \left(G_1 FINT + G_2 DBLINT\right) * G_3 \tag{4.9}$$

where CS, FINT and DBLINT are the control signal, the filtered integral and double integral terms, respectively. The term G_3 was used as a global system gain.

After each time step is solved for, the single time post-processor was entered to obtain the voltages from the sensor piezoelectric element using the *GET command. The control law finite difference algorithm was performed and then applied as a voltage to the actuator element in the next time step. The control signals and the tip displacement were captured as well, for later analysis.

The variable definition, load establishment, transient loop and control law integration are all included in the APDL macro "Truss.inp." The ".inp" suffix was used to indicate that the file is used as an input file, and carries no special meaning with ANSYS. "Truss.inp" is included as Appendix G.

4. Actively Controlled Model

With the pieces assembled, namely the model and APDL macro program, "Truss.inp," controlled runs to evaluate effectiveness of the control parameters could be performed. Initial trial runs of the model revealed that the initial start of the applied force created a transient response that operated at the natural frequency of the model. To

remove this effect and allow for examination of control of the driven frequency, damping was added to the model.

ANSYS has three different damping options available, constant, Rayleigh and modal. Not all options are available with the different analysis types. For the transient analysis, only Rayleigh damping is available. Rayleigh damping creates a [C] matrix by using constants multiplied by the Mass and Stiffness matrices. The damping matrix is given by the following equation:

$$[C] = \alpha[M] + \beta[K] \tag{4.10}$$

Rayleigh damping can be used to approximate the structure in the frequency region of interest, given the damping ratios for the given modes. The damping ratio, ζ, is a function of the Rayleigh damping coefficients based upon the following equation:

$$\zeta_i = \frac{\alpha}{2\omega_i} + \frac{\beta\omega_i}{2}. \tag{4.11}$$

A set of simultaneous equations for two frequencies can be set up, and then the Rayleigh constants may be solved. For this analysis, the damping coefficients set as 10-percent, giving α to be 0.085 and β to be 6.1e-4. The 10-percent value was arbitrary and was selected to damp out the transient vibrations.

The time step was selected using the guidance presented in section 5.12 of Reference 26, the ANSYS Structural Analysis Guide. Under this guidance, a time step that was one-twentieth the applied frequency was used. This provided results that were accurate enough for this qualitative analysis. As discussed earlier, there is not an appreciable error above five-percent of the sampling rate. Sampling at one-twentieth the applied frequency will place the driving frequency at five-percent of the sampling rate.

The Piezo5 model was run under uncontrolled conditions and with varying G_3 settings. The model was excited for 2.0 seconds at 16.75 Hz, the frequency used by Vlattas and Johnson [Ref. 14]. The 2.0-second run time was found to minimize the processing time while giving ample data for frequency analysis. After initial trial and error, it was discovered that a polarity mismatch existed in the actuator. To correct this, gains of negative value were used. This solved the problem and enabled for reducing the amplitude of the tip motion. It was found to be more efficient to clear the database and start fresh prior to a new run to prevent corrupt constants from affecting the results.

Nine data runs were performed on the Piezo5 model. The first was uncontrolled, and the rest varied the G3 system gain term from 0.0 to –20.0. The results are presented below.

Run	G3	Amplitude	.dB.
1	0	1.670E-07	0.000
2	-5	1.560E-07	-0.592
3	-10	1.497E-07	-0.950
4	-15	1.507E-07	-0.892
5	-20	1.552E-07	-0.636
6	-8	1.530E-07	-0.761
7	-12	1.491E-07	-0.985
8	-13	1.497E-07	-0.950
9	-11	1.486E-07	-1.014

Table 4. Piezo5 Active Control Results

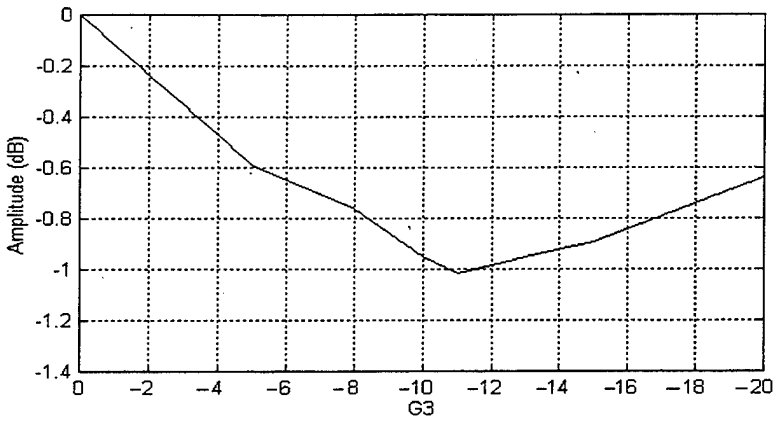


Figure 21. Piezo5 Active Control Results

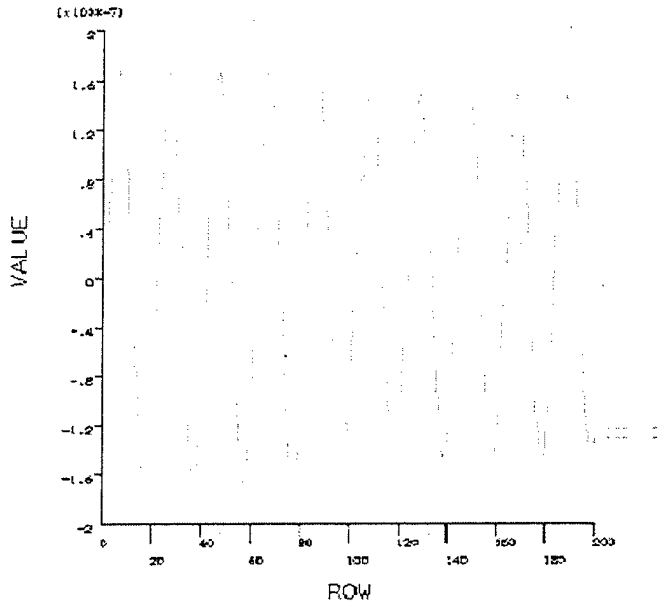


Figure 22. Tip motion of Piezo5

Figure 22 shows the motion of the tip of the Piezo5 model when the active control is activated. The x-axis of the plot reflects the time index, each number represents a sample step; the active control is turned on at step 100. The amplitude is small, on the order of 10^{-7}. As can be seen in the above results, the best case for the actively controlled model occurred when the G3 term was equal to –11.0 with a reduction in magnitude of –1.014 dB. The amplitudes are small, and the results are modest, but this is consistent

with the fact that Piezo5 was intended as a demonstration model to prove the controllability of the FEM approach.

C. NPS SPACE TRUSS

1. Model Construction

The first step in the ANSYS analysis of the NPS was to create a bare truss FEM. A simple structure, such as the NPS space truss can be rapidly constructed by using some of the features inherent to ANSYS. The WorkPlane is a user modifiable plane that allows the user to select points and draw on the model on the plane surface. In this manner, when a point is selected on the screen it is located where that point intersects the WorkPlane in 3-D. The WorkPlane grid was set up such that the spacing between grid points was the distance between nodes on the actual truss. Keypoints were located at these points by snapping them to the grid and joined by lines in same manner as the truss. The elements used to mesh the model were BEAM4s and MASS21s. The BEAM4 elements used the geometry of the aluminum truss elements. The MASS21 elements were used to account for the additional mass of the fittings, connections, node balls and the knurled knobs that were attached to the node balls.

2. Bare Truss Modes and Natural Frequencies

To verify the FEM created with ANSYS, a modal analysis was performed. When an analysis is performed on ANSYS, the program will deliver a summary of mass properties and CPU allocation parameters. The mass of the model was compared with the mass tabulated in Appendix A. The mass of the FEM of the truss was 11.689 kg, as

compared to 11.708 kg for the actual truss. ANSYS was directed to provide the first 10 natural frequencies and the first four mode shapes.

The mode shapes obtained from ANSYS for the bare truss model are presented in Table 5. The frequency sequence follows the order and sequence of those obtained from the MATLAB FEM presented in Chapter II, and the experimental results obtained previously [Ref. 14].

ANSYS will provide to the user, if requested, various plots and tables of the results. The mode shapes for the first four modes were obtained in this manner. It was found that these mode shapes were identical to those obtained with the MATLAB FEM earlier.

Mode	Frequency	Mode	Frequency
1	14.249	6	72.932
2	15.567	7	81.674
3	28.928	8	96.608
4	32.261	9	115.416
5	60.761	10	122.261

Table 5. Natural Frequencies of ANSYS Bare Truss FEM

D. INTEGRATION OF THE ACTIVE CONTROL ELEMENT

The steps necessary to create an active finite element in a preexisting structure follow a similar pattern to the steps to create the sample model, Piezo5. In a realistic model, geometry and material constants become important. These step are laid out as follows:

- Unmesh the FEM in the location of the active element.
- Create geometry for the active elements: Lines for the connecting parts and volumes for the sensor and actuator.

- Create the material and mechanical properties for the connecting parts and the piezoelectric materials.
- Orient the WorkPlane to align the 3-axis with the axial direction of the active elements and create a local coordinate system fixed to the WorkPlane origin.
- Create Mesh: BEAM4 elements for the support connections, SOLID5 elements for the piezoelectric elements. Mesh volumes as a single SOLID5 element. Ensure that SOLID5 elements have the local coordinate system as the elemental coordinate system.
- Create rigid region (in displacement DOF) to couple the ends of the sensor and actuator to the connecting pieces.
- Apply fixed voltage DOF at one end of the sensor and actuator elements.
- Couple the other end of the active elements in the voltage DOF.
- Create the SOLID5 data tables for the piezoelectric properties (PIEZ).

The NPS space truss active FEM has two different piezoelectric materials; the force transducer is quartz and the actuator is made of PZT. The constants needed for this analysis are the axial stiffnesses, piezoelectric constants and dielectric constants. The governing equations that ANSYS uses for the piezoelectric effects are in the stress form:

$$\begin{Bmatrix} T \\ \vec{D} \end{Bmatrix} = \begin{bmatrix} [c] & [e] \\ [e]^T & -[\varepsilon^{\epsilon}] \end{bmatrix} \begin{Bmatrix} S \\ -\vec{E} \end{Bmatrix}. \tag{4.12}$$

For piezoelectric properties, ANSYS required the [e] matrix as opposed to the [d] matrix. By convention, the piezoelectric coefficients are given to populate the [d] matrix, and must therefore be converted to the [e] matrix by the following equation:

$$[e] = [c]^{\epsilon}[d]. \tag{4.13}$$

Since the piezoelectric materials are only constrained in the axial (or 3) direction, some simplifications to the constant determination process may be used. Only the 33 terms of the [d] and [e] matrices are important. By omitting the off-poling axis terms from the PIEZ data tables, the piezoelectric effect is therefore nullified, and the constraint that was presented earlier with the rigidization of the ends is alleviated. Therefore, Equation (4.13) can be simplified to

$$e_{33} = E_{33} d_{33}. \tag{4.14}$$

For the quartz force transducer, the material specifications provided the sensitivity and stiffness (see Appendix A for the full component specifications). By using the stiffness and sensitivity, the d_{33} and E_{33} terms may be determined by direct combination. The elastic modulus was derived from the stiffness using the following equation:

$$k = EA / L. \tag{4.15}$$

The size used for the quartz element was a cube, 1.0 cm per side.

The d33 constant for the actuator was obtained from the experimental work performed by Johnson and Vlattas in Reference 14. As the macroscopic properties were determined experimentally, it was possible to use a single element to model the actuator, even though it was composed of several PZT wafers, electrically connected.

A summary of the material properties used for the SOLID5 piezoelectric elements is presented in Table 6. SOLID5 Material Properties.

Property	Quartz	PZT
Stiffness	1.0 kN/μm	33.0N/μm
Sensitivity	11,420 mV/kN	-
Axial Elastic Modulus	100Gpa	142GPa
d_{33}	87.56e-9 m/V	500e-9 m/V
e_{33}	8756	7100

Table 6. SOLID5 Material Properties

A visual depiction of the piezoelectric sensor and actuator is shown in Figure 23. This figure was obtained from a hard copy screen capture from ANSYS.

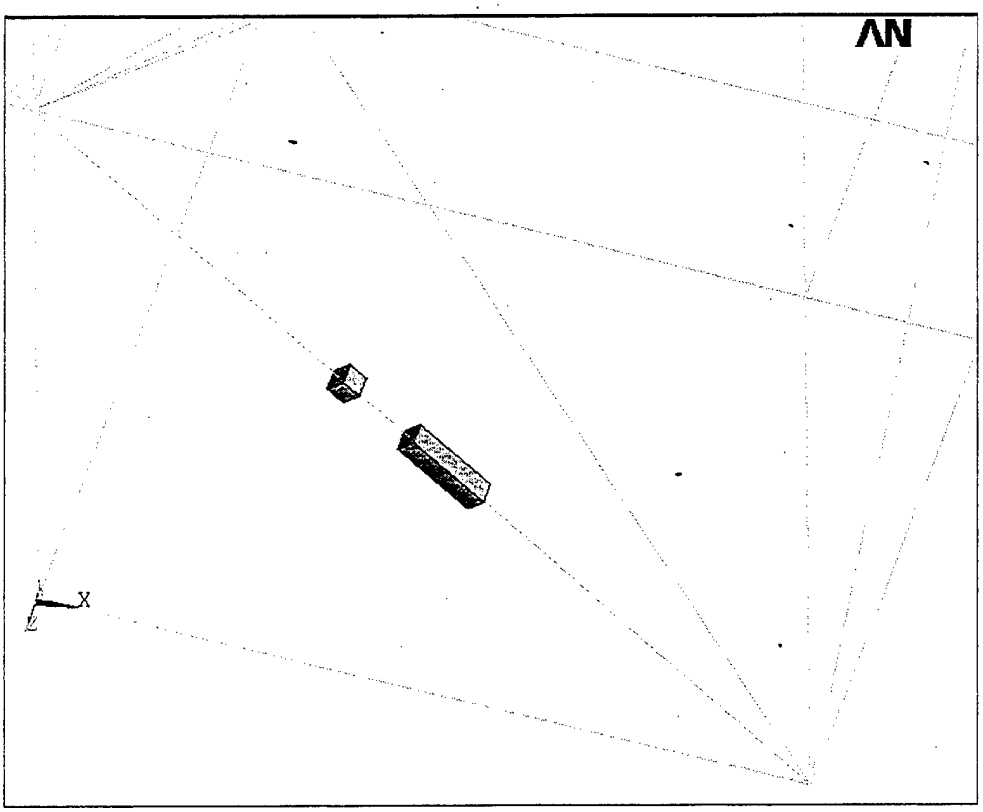


Figure 23. Installation of Active Members in NPS Space Truss

The LPACT was modeled by creating a MASS21 element with the mass of the device at the appropriate location. The transfer function was used to determine the applied force. As was shown in Chapter II, the LPACT has a natural frequency at approximately 8-10 Hz. This location is less that the desired frequency range for the

truss, but as can be seen from Figure 11, there is large variation in the transfer function from 10 to 20 Hz. Using picked data points and the MATLAB function "Polyfit.m" a quadratic transfer function was derived. This function is given by the following equation:

$$AMP = 0.2037 * FREQ^2 - 7.0719 * FREQ + 68.3564 \text{ (lb/amp)} \qquad (4.16)$$

Using unit conversions and the electrical characteristics for the LPACT, a function could be applied in the macro APDL program that would simulate the change in the LPACT force that would be applied in the FEM. The element that contained the LPACT was divided into two separate elements to create the location for the MASS21 element and the location for the force application.

The commands that were used to create the finite element in the NPS space truss were recovered from the log file. An edited version of these commands is included as Appendix H. Commands that controlled the display were among those omitted from the appendix.

For the NPS truss active FEM, Rayleigh damping was again used. Given the limitations of modeling Rayleigh damping for a structure with a large number of DOF, it was decided to select a value of 10-percent for the damping ratio, at the first two natural frequencies. Using the first two natural frequencies for the actual truss with the components installed, 13.113 and 17.125Hz, the Rayleigh constants, α and β, were determined to be 9.1 and 1.1×10^{-3} respectively. The 10-percent value was an average of experimentally determined values in the band of interest [Ref. 14, p. 111]. Equation (4.11) was used in a series of simultaneous equations to perform this calculation.

The APDL macro program, "Truss.inp" was modified to become "Act_truss.inp", which served as the root program for the actively controlled NPS space truss model. The modifications include changing the node locations to the appropriate nodes in the truss model, and adapting the routine for the LPACT forcing function. The 20-to-1 gain provided by the Trek 50/750 Voltage amplifier was lumped into the SG term. "Act_truss.inp" is included as Appendix I.

The first test runs of the active NPS space truss FEM revealed a significant error in ANSYS. Once the model was complete, including the PIEZ tables, a modal analysis was attempted to determine the new natural frequencies, for comparison with the experimental data. ANSYS version 5.5 does not permit a modal analysis to be run with the PIEZ tables in the model. This path was abandoned and the transient analysis was attempted. Upon the attempt of the transient analysis an error was produced that stated:

> "Piezoelectric materials may not be used with the subspace eigensolver. Please use the reduced solver (MODOPT,REDUC command) with master DOF's only at the displacement nodes."

The subspace solver is an algorithm used in modal analyses to speed up the calculations, and was used for the both the bare truss model and the attempt at the active model. When the transient analysis mode is selected, ANSYS does not even permit the de-selection of the subspace solver. The problem was eventually resolved by exploring the model's data base file using the command

Utility Menu>File>Write DB Log File LGWRITE

The LGWRITE command revealed that an artifact of the modal analysis run on the bare truss model was still present in the data base file. This was resolved by editing

the data base file to remove the references to previous modal analyses. This error has been brought to the attention of ANSYS, Inc and hopefully will be resolved in future versions. [Ref. 27]

For the larger NPS space truss model, selection of a time step seemed to be more crucial due to the larger number of system degrees of freedom. A poorly selected time step can cause some numerical instability in a transient analysis [Ref. 15, p. 285]. Include the running of the model with negligible force. Early runs of the repaired model proved to be unstable after the third or fourth time step. By changing the sampling frequency, this problem was unable to be alleviated. Only by running with infinitesimal amplitudes could the model be run.

This was later determined to be from incomplete data in one of the data tables. Initially, the ANEL data table was used to define the anisotropic material properties. This table had incomplete data and was causing large shear stresses to exist in the model during transient analysis. By removing the ANEL data table, and placing the elastic constants in an orthotropic state, this problem was alleviated, allowing the FEM to run through its specified time. [Ref. 28]

The model was reconstructed from a version that was incomplete and had never had an analysis performed on it. After the bare model was reconstructed, a transient analysis was performed on the bare model using a version of the active APDL macro with the active parts removed by commenting. The active components were installed with the exception of the PIEZ data tables. The transient analysis was run successfully on the model. The PIEZ data table and material properties were created and the model was able

to run without instability. Further evaluation of the NPS space truss FEM is contained in the next chapter.

E. ALTERNATE SIMULINK MODEL

In order to evaluate an active piezoelectric control element in a Simulink analytical model, its operating characteristics must be modeled. The derivation of this modeled transfer function, from force sensor to piezoelectric actuator is complex, and will be explained in detail.

For this model, built from the independent FEM used in Chapter II, it was assumed that the piezoelectric element and the strut are the same material and length and co-located. This is not realistic for a quantitative model, but it is sufficient to prove controllability of this model. The embedded force sensor detects motion from the axial expansion or contraction of the element. The nodal displacements must be converted to the local frame by

$$\{x\}^l = [DCM]^{lg}\{x\}^g. \tag{4.17}$$

For a truss element, only the axial change in length is important. This change is given by:

$$\delta^l_{axial} = u_2 - u_1 = [-1 \quad 0 \quad \cdots \quad 1 \quad 0 \quad \cdots]\{x\}^l. \tag{4.18}$$

Using the stress-strain relation gives

$$F_{axial} = k_{effective}\delta^l. \tag{4.19}$$

The active element can be modeled as springs in series; the effective stiffness of the element is therefore the reciprocal of the sum of the reciprocals, given by the following:

$$k_{effective} = \left(\sum_{i}^{NE} \frac{1}{k_i} \right)^{-1} \tag{4.20}$$

From Newton's first law, the force in the sensor is the same as felt by the member. Therefore, the change in length for the sensor is given by

$$\delta_{sensor} = \frac{F_{axial}}{k_{sensor}}. \tag{4.21}$$

The signal voltage output from the sensor is given by the following equation

$$V_{sensor} = \delta_{sensor} / d_{33}. \tag{4.22}$$

The control law used, as described previously, is presented below

$$C = \left(\frac{G_1}{s} + \frac{G_2}{s^2} \right) V_{sensor} K_1 \tag{4.23}$$

where C is the control signal K_1 is a constant term reflecting the gain from an amplifier on the output of the force sensor to the processing computer.

The output from the controller is applied through a voltage gain amplifier and is applied through the piezoelectric effect to the long axis of the element as a control force. This relation is given by

$$\{F_{control}\}^l = \begin{Bmatrix} -1 \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \end{Bmatrix} k_{piezo} d_{33} K_2 C, \tag{4.24}$$

where K_2 is the amplification gain from a voltage amplifier and the computer output gain. The voltage limiter in the dSPACE model was not included in this model to simplify its construction.

Converting to the global coordinate frame from the local frame gives

$$\{F_{control}\}^g = [DCM]^{gl}\{F_{control}\}^l. \tag{4.25}$$

Combining the above equations yields

$$\{F_{control}\}^g = [DCM]^{gl}\begin{Bmatrix}-1\\0\\\vdots\\1\\0\\\vdots\end{Bmatrix}K_2\left(\frac{G_1}{s}+\frac{G_2}{s^2}\right)K_1\frac{k_{piezo}}{k_{sensor}}\frac{d_{33\,piezo}}{d_{33\,sensor}}[-1 \quad 0 \quad \cdots \quad 1 \quad 0 \quad \cdots][DCM]^{lg}\{x\}^g \tag{4.26}$$

Equation (4.26) is cumbersome; as most terms are constants, this equation can be reduced to

$$\{F_{control}\}^g = [AP]K\left(\frac{G_1}{s}+\frac{G_2}{s^2}\right)[ISO]\{x\}^g \tag{4.27}$$

where [ISO] is the vector that isolates the desired displacements from the truss converts them to the local frame and determines the change in element length. [AP] is the application vector that takes the axial output of the active strut and applies it to the truss structure. For this analysis, the exact value of the constants is unimportant, and served only as an alternative exercise for the author. Therefore, all the constant terms were lumped together as the variable K. This allows the computer to perform the integral and double integral transfer functions on a single scalar, as opposed to the entire state vector.

The active control model was built in Simulink, using 300 for G_1 and 100 for G_2. This was in keeping with the results from Reference 14. A twenty-second run was performed on each location, acting on an impulse disturbance acting at node three in the y, z and torsional x-directions. This three-degree of freedom impulse was selected to

excite several different vibrational modes of the truss. For consistency, for each run, the output was obtained from the y-displacement of node 3, and a lumped gain, K, of 30,000.

A time step of 0.0001 sec was used, as larger time values were found to be numerically unstable in the model. Kwon [Ref. 15, p. 285] states that the time step must be less than the minimum model period divided by π. An analysis of a desired time step based upon the maximum frequency of interest criteria was not performed as stability was the only desired objective. Any attempt at qualitative analysis should include an appropriate time-step, based upon either frequency of other appropriate criteria.

Early evaluation revealed a bias in the values for sensed displacement, similar to the experimental results obtained earlier. This is believed to be an artifact of the numerical analysis, and was alleviated by inserting a Butterworth high-pass filter in the control stream with a high-pass frequency of 3 radians/second. The complete Simulink model for this analysis is shown in Figure 24.

In order to prepare the FEM for the Simulink model, the program "NPS_prep.m," included as Appendix J, was created. "NPS_prep.m" served to perform the matrix inversions and multiplications required and to create the [AP] and [ISO] matrices.

The model produced a qualitative validation of its controllability. In the test runs, the control law was activated at 5.0 seconds by changing the lumped gain value from zero to 30,000. A screen capture of the sensed control input is shown in Figure 25.

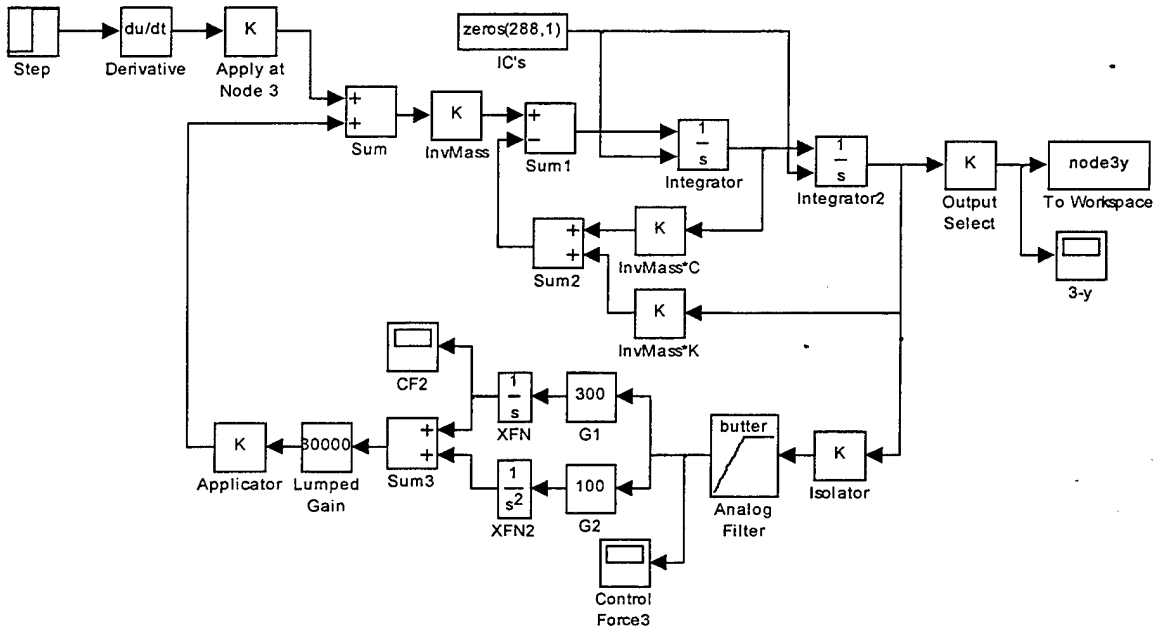


Figure 24. Simulink Active Controlled Model

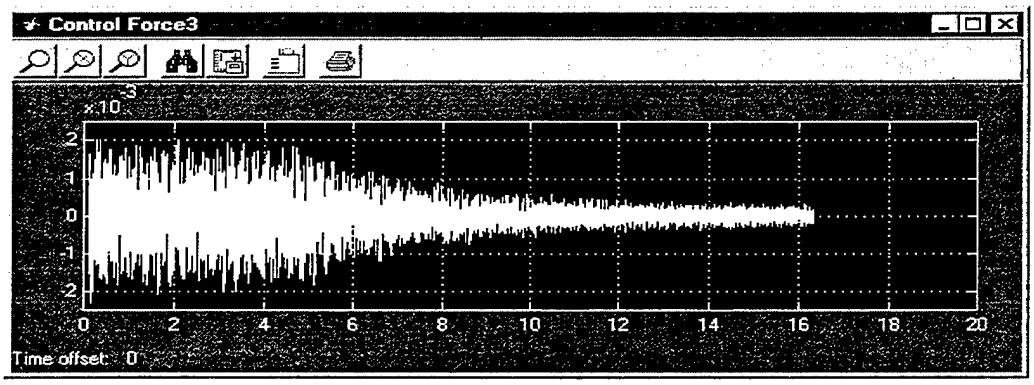


Figure 25. Sensed Signal From Active Simulink FEM

V. COMPARISON OF ANSYS FINITE ELEMENT MODEL TO NAVAL POSTGRADUATE SCHOOL SPACE TRUSS

A. METHODOLOGY

The purpose of the experiments performed in this thesis was to verify the active FEM developed in the previous chapter. To do this, five experiments were performed on the NPS space truss to evaluate the IDIFF controller's performance.

Before performing control parameter experiments, the truss was excited with random noise from a digital spectrum analyzer to verify the natural frequencies. The first experiment that was performed was an evaluation of the controllability of the first natural frequency. The second natural frequency was targeted for the next experiment. For the third experiment, a repeat of the experiment performed by Johnson and Vlattas [Ref. 14] was done. The fourth experiment was to combine the first two natural frequencies with variable phasing. Finally, random noise was applied and controlled with the IDIFF controller.

For each experiment, a series of runs were performed with varying gain settings. The value of the read sensor was used to help find the optimum point. The goal in these experiments was to reduce the vibration on node 26, simulating the location of a sensitive piece of equipment. The intent of the LPACT was to provide excitation from a simulated vibrating component operating at critical frequencies.

B. EXPERIMENTAL CONTROL SYSTEM

1. Experimental Setup

In order to detect the motion of the truss, four Kistler 8690C10 three-axis accelerometers were used. Two of the accelerometers were place at the extremities of the truss, nodes 26 and 41. FEM modal analysis revealed that these locations experienced the largest motion when excited at the natural frequencies. These tip sensors also will capture the motion from the first several modes. The second pair of accelerometers were placed at the midpoint of the truss arms, on opposing sides to the tip accelerometers (nodes 49 and 18) to capture the higher modes where the ends of the truss do not vibrate as much as the middle.

The output of the accelerometers was fed into a Kistler Piezotron Signal Conditioner, model 5124A. The signal conditioner does not amplify the signal, and only served to power the accelerometers and filter the output. The accelerometers experienced significant drift during the course of the experiments. This drift was able to be easily isolated from the signal of interest with a spectrum analysis.

The heart of the digital control system was the DS1103 digital signal processor from dSPACE. The dSPACE system received inputs from the quartz force transducer, after conditioning to perform its control function. The conditioned accelerometer signals were fed into the ADC inputs for structural monitoring.

To perform a pseudo-integrated monitoring function a Hewlett-Packard HP 54601A four-channel oscilloscope was used to monitor the read control signal from the force transducer and the output signal to the piezoelectric actuator. The oscilloscope was

set up such that both channels could be monitored at a sufficient resolution to obtain a rapid qualitative feel for the behavior of the system at a glance.

To supply the excitation to the LPACT, a Hewlett Packard HP 33120 digital signal generator as used. The HP 33120 has the capability to provide many different wave shapes, at a variety of frequencies and amplitudes. Selecting a frequency and amplitude can be performed by direct keypad insertion at the front of the signal analyzer, or by using a dial to scroll through the selected significant digit on the display.

All the connections between components were provided y varying length BNC connector cables. These cables had no special specifications. The completed experimental control system can be seen in Figure 26. Experimental Layout, shown below. A detailed description of the cable connections is provided in Table 7.

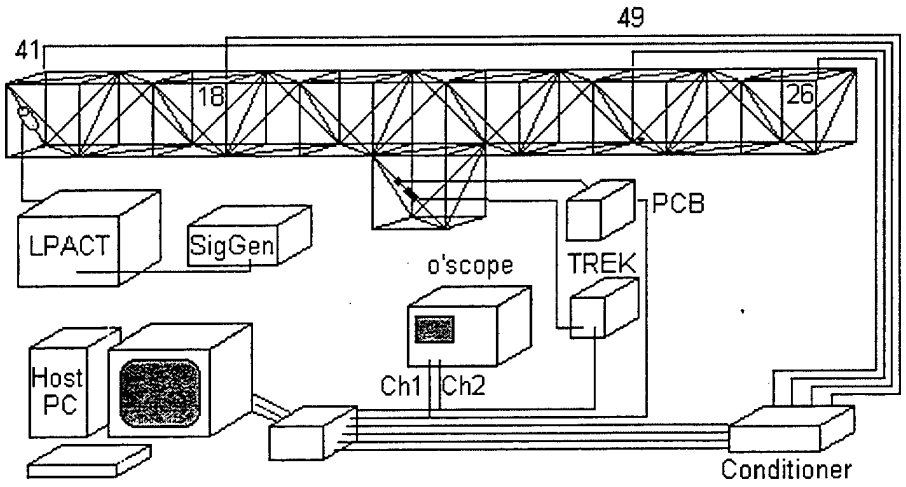


Figure 26. Experimental Layout

From Device	Connection	To Device	Connection
HP 33120A	Output	LPACT Driver	Current Command
Accelerometers	Cable	Kistler Coupler 5124A	Input
Kistler Coupler 5124A	Output	dSPACE I/O Box	ADCH 1-3,5-7,9-11,13-15
PCB Force Transducer	Cable	PCB Model 484B	XDCR
PCB Model 484B	Scope	dSPACE I/O Box	ADCH 17
dSPACE I/O Box	DACH 1	TREK 50/750	INPUT
TREK 50/750	OUTPUT	PI Piezo Actuator	Cable
dSPACE I/O Box	DACH 1	HP 54601A "scope"	1
PCB Model 484B	Scope	HP 54601A "scope"	2

Table 7. Experimental Cable Connections

Once the cables are installed, the truss may be activated. The first step in activating the experiment is to power up all the equipment that will be used. Finally, the Newport Vibration Isolation table will need to be floated by opening the valve to the Nitrogen tank. Over time, the compressed gas will bleed from the pneumatic cylinders supporting the table; therefore, care must be taken to ensure that the valve is shut upon completion of the experimental data taking, to prevent the exhaustion of the Nitrogen supply. As the experiment is located in the basement of Halligan Hall, this evolution takes some time to perform, and draws away from the time available for experimentation.

The dSPACE ControlDesk program is then called up on the host PC, using the steps described in Chapter III. To summarize, the ControlDesk experiment "ST_controller" is opened, which will open the panel layout file and the trace variable file. Finally, the "ST_controller_3" program is loaded into the CPU.

At this time the signal generator may be activated and tuned to the desired frequency and amplitude. The LPACT does not activate until the LPACT switch is placed to "enable." When enabling the LPACT, caution must be taken to ensure that the actuator is being held secured, this will prevent a large transient from going through the

truss and damaging sensors due to a large response. Also, when active, adjust the applied voltage slowly using the vernier dial on the front of the signal generator, to prevent the same transient. The frequency may be adjusted by the use of the keypad while the LPACT is enabled, as this does not cause a noticeable transient to occur.

2. Modal Verification

Before the experiments were performed to evaluate the active control system with the IDIFF controller, the modal frequencies were verified with a Hewlett-Packard HP35665A Dynamic Signal Analyzer (DSA). The HP35665A has to ability to generate a random noise signal within a desired frequency range and can analytically isolate the desired frequency range. This analysis was also performed to determine the highest effective frequency, which was used to obtain the time step size for the Simulink model.

The DSA was set up in two-channel mode, with a split screen to allow viewing of both channels. Each channel was set up for fast Fourier Transform (FFT) analysis. One channel was set to record magnitude; the other channel was set for phase measurement. The modes become apparent where there are local peaks in magnitude with a corresponding 180-degree shift in the phase. As random noise was used to obtain the frequency response, averaging was used to smooth the signal. The averaging method used for this analysis was vector averaging.

Vector averaging preserves the complex values of the signal, and allows averaging phased data, such as an FFT analysis [Ref. 27, p. 4-313]. The phase information in the vicinity of the modes was still degraded due to the rapid phase shift at these locations. By looking at the data in mid-average, it was easier to determine where

the modes were. The HP 35665A DSA setup procedure is included as Appendix K. HP 35665A DYNAMIC SIGNAL ANALYZER SETUP. The equipment setup is shown in Figure 27. The random noise source, in addition to being the driver for the truss, was fed back into channel one of the DSA as transfer function input. The response from the force transducer was sent to channel two as transfer function output.

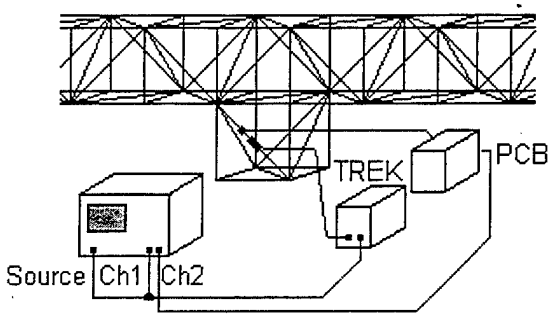


Figure 27. HP 35665A Modal Determination Setup

The DSA is also capable of saving the data on a 3-1/2 inch floppy disk. These data files are saved in a format called SDF (Standard Data Format). A routine was provided with the HP 35665A DSA to perform a conversion to standard MATLAB MAT-files. This routine is called SDFTOML. This routine is executed from an MS-DOS window with the following syntax:

MS-DOS Prompt> SDFTOML <SDF filename.EXT> <MAT-filename.MAT>

The data that is recorded in the MAT-file consists of four variables, o2i1, o2i1x0, o2i1xi, and o2i1xl. These variables are the data, which is a matrix of dimension two by the number of resolution lines, plus one, and the frequency specifications. The variables o2i1x0, o2i1xi and o2i1xl are the starting frequency, the frequency resolution and the ending frequency, respectively.

The captured data was transferred to the dSPACE host PC in the sub-directory: C:\Space_Truss\truss99\experiments\HP_DSA. Six runs were performed, examining the frequency at varying resolutions over zero to 200 Hz. By examining smaller frequency ranges, a greater resolution was achievable. The data files were examined with the M-file DSA_PLOT.M, included as Appendix L. The range from zero to 50 Hz is presented as Figure 28.

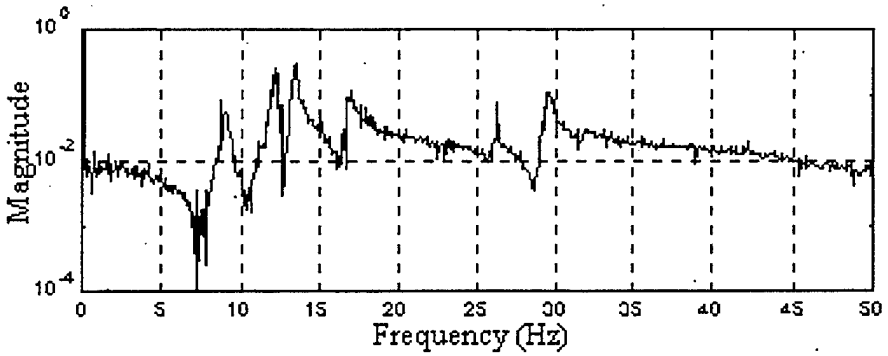


Figure 28. Frequency Response from Random Noise Response (0-50Hz)

The first and second natural frequencies will be targeted for this thesis, as well as the 90-degree offset point examined by LT Vlattas and LT Johnson [Ref. 12]. Due to the resolution of the DSA, the precise location of the natural frequency was not obtained, but a small band was used with the vernier feature of the signal generator to lock in on the natural frequencies.

As can be seen from the results there is a new natural frequency in the 8-10 Hz range. This frequency is due to the addition of the LPACT to the truss structure. This natural frequency is local to the LPACT strut, given the low stiffness of the LPACT drive spring, and was therefore not included as part of the frequency range of interest. This was confirmed by examining the first mode shape from the truss with the active components installed, as generated in ANSYS.

The frequency range of zero to 200 Hz was examined in order to determine the maximum frequency of interest for the truss. As can be seen from Figure 29, the magnitude of the frequency response begins to taper off at 200 Hz. For this reason, frequencies above 250 Hz, or sampling at 500 Hz, should provide for negligible aliasing effects.

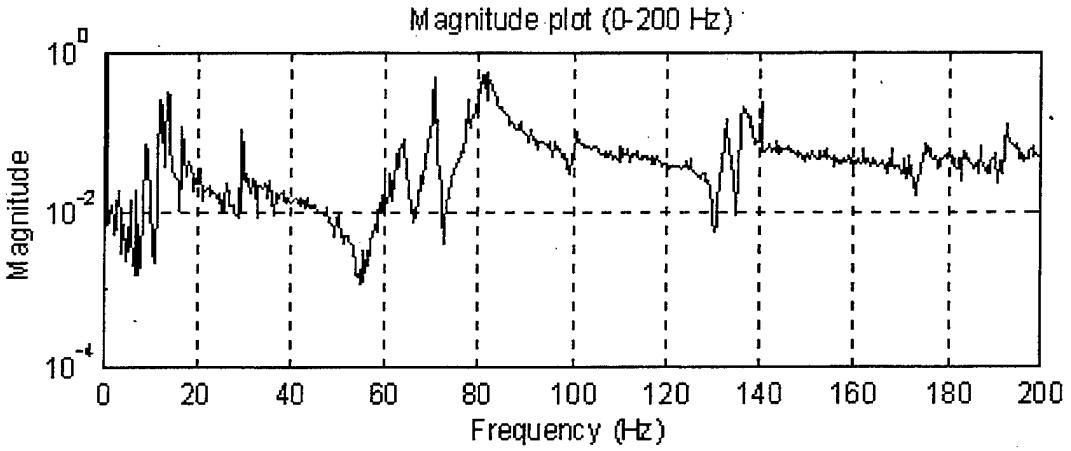


Figure 29. Frequency Response from Random Noise Response (0-200Hz)

C. EXPERIMENTAL PARAMETERS AND RESULTS

1. Data Capture

The dSPACE software package also includes two function libraries that allow real-time access to the DS1103's memory. These two utilities are called MLIB and MTRACE. These functions were designed to be executed from within the MATLAB workspace, or from m-files that the MATLAB program is running. MLIB provides access to the parameters that the CPU is using for the running dSPACE application, MTRACE provides for continuous or timed data capture capability [Ref. 30, p. 6]. A pictorial description of this relationship is shown in Figure 30.

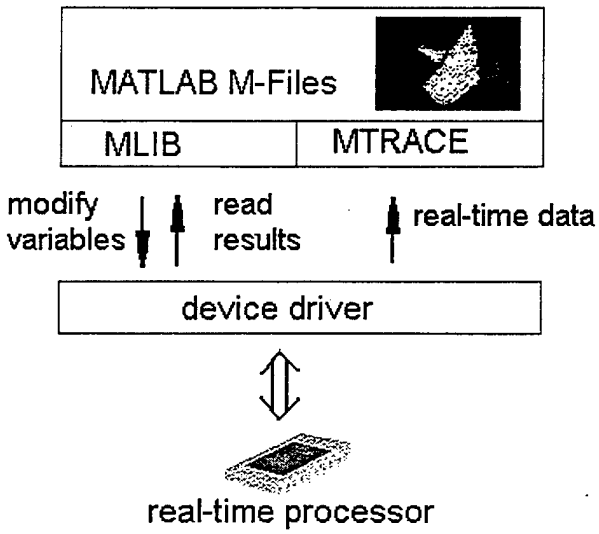


Figure 30. MLIB/MTRACE Relationship [From Ref. 30, p. 6]

The MLIB and MTRACE routines are invoked by function calls from within MATLAB m-files. The specific function desired is called as a parameter to the function call, with the details of the command being secondary parameters. Examples of this are:

```
mlib('DesiredCommand','parameter_name','parameter_value','etc.');
mtrc1103('DesiredCommand','parameter_name','parameter_value','etc.');
```

The MLIB and MTRACE libraries were used to instruct the DS1103 CPU to provide 20 seconds of data for each experimental run, at the Simulink model-sampling rate of 500 Hz, for a total of 1,000 samples. The parameters that were obtained were the accelerometer readings, the force transducer input and the dSPACE signal output to the piezoelectric actuator. The program "Acq_data.M," included as Appendix M, contains the code that was used. "Acq_data" was modified for each experiment to ensure that the data obtained would be deposited in the correct subdirectory.

Using the guidance contained in References 14 and 30, the program was instructed to capture and process the data in the following manner:

- The CPU was polled to verify that "ST_Controller_3.ppc" was running.

- Variables were defined and dSPACE instructed to find their memory locations.
- Capture settings were established, to include sample length. Data can either be stored on the DS1103 local memory or sent to the host PC for storage on the hard drive.
- After the data capture was complete, it was sent to the MATLAB workspace as a matrix of dimension, number of variables by number of samples.
- The data were segregated into individual vectors for plotting and performing frequency analysis.

One of the improvements in the ControlDesk program was the inclusion of a D&D function that can retrieve the variable name directly into the MATLAB editor for inclusion in an M-file. When a Simulink model is created and sent to the CPU with the RTI program, it creates a trace file that contains the memory locations and names of the model parameters. These parameters are sorted by dSPACE by function and block location, as described in Chapter III. The D&D function saves time by alleviating the need for the programmer to re-type the entire variable name, as seen by dSPACE, and reduces programming errors.

After the first run for each experiment, which was uncontrolled, the program was altered to use the first set of data as a comparison. All the captured data was saved as a separate MAT-file for later processing. The data processing routine is included as "Data_proc.m," Appendix N.

2. Experiment 1, First Natural Frequency, 12.50 Hz

The first experiment performed was a control of the first fundamental frequency of 12.50 Hz. This frequency was verified prior to experimental runs by using the vernier dial on the HP 33120A Signal Generator. The vernier dial was adjusted until the largest response was seen on the oscilloscope display. The amplitude on the signal generator was varied until a noticeable and clearly sinusoidal response was seen on the oscilloscope. An amplitude of at least 1.0 Volts peak-to-peak (Vpp) was required to obtain this response.

Raising the voltage to 1.2 Vpp allowed the laser diode to provide a clear visually qualitative picture of the controllability of the truss. The laser diode reflected the side to side nature of the mode shape by its projection on the laboratory wall.

The first set of runs was performed in phases. First, the single integral term, IGain, was set at 300 and the system gain, SG, was modified to find the limit of stability. Once the limit was reached, SG was backed off slightly to allow for variation in IIGain, the double integral term, to find an optimum point. This was then repeated with IIGain being the dominant variable. Finally, the amplitude was lowered to attempt to determine if the 100 mVpp signal reported in Reference 14 was possible.

For each run, the amplitude and power reduction from the uncontrolled truss at node 26 in the Y-direction were determined. Node 26 is at the opposite extreme from the truss as the LPACT. These values are listed for all the data runs in Appendix O.

In order to conserve space, the graphical plots of the captured information (control signal input, dSPACE control output and the three-axis response of the four accelerometers), presented in Appendix O, will represent a snapshot of the runs from all

experiments, showing a generic uncontrolled case, unstable cases that are borderline stable and rapidly unstable and the best case controlled case. These plots are presented as Figure 46 through Figure 56.

During the test runs, it was discovered that the Nitrogen had run out of the bottle, requiring a replacement. The next few data runs were slightly inconsistent with the first set. Due to the somewhat random nature of the data, there is some scatter from one measurement to the next, and from day to day. In order to obtain a consistent series of results, an experiment should be completed as soon as possible, within the same day, or hour as the data runs are only 20-seconds long.

When instability occurred, the active strut traveled from saturation on one side to saturation on the other at a frequency that is different that any of the truss' natural frequencies. High frequency transients are observed as the actuator enters and exits the saturated region. The spectral results show this transient as a series of spikes on the PSD plot. Also observed was accelerometer drift that occurred during the experimental runs. This was a low frequency effect and was apparent on the spectral analysis, but appeared in the lower frequency band (<2 Hz), and therefore was not of further interest.

In all, 28 trials were performed, in addition to the uncontrolled case. The best results were obtained from trial 15, with parameters of IGain=300, IIGain=-400 and SG=1.75. The reduction in magnitude from this trial was 18.54 dB.

The relative contribution of the IIGain parameter was two orders of magnitude less than the single integral term. This relationship can be simplified by the following equation:

$$\int A\cos\omega t dt = \frac{A}{\omega}\sin\omega t \tag{5.1}$$

It was discovered that the double integral term, in order to be amplified to a level where its effect could be felt on the truss, drove the system more rapidly into instability by amplifying the noise as well. When used in conjunction with the IGain term, the IIGain term can be most effectively used to adjust the frequency response when damping out the disturbance. The IGain term refers to rate feedback, and affects the damping; the IIGain refers to position feedback, and modifies the system natural frequencies.

Of note from the results is that as the desired location is reduced in amplitude, the other locations monitored show an increase. This is due to the installation of only one control strut in the truss. In order to target multiple degrees of freedom, and different mode shapes, more control struts would be required. This would also boost the control authority and further reduce the magnitude. Care must be take in selecting a second strut location to ensure that the desired effect is obtained.

If multiple modes are targeted from multiple control struts, a method must be developed to assist with the selection of the active strut locations. One method is to use a weighing of the strain energies (in a manner similar to that in Chapter I, but for multiple modes) to select the optimum locations. This method was used successfully in previous research to control a cantilevered truss [Ref. 31, p. 394].

3. Experiment 2, Second Natural Frequency, 13.81 Hz

The second set of experimental runs was targeted at the NPS space truss' second natural frequency, 13.81 Hz. As with Experiment 1, this value was obtained by adjusting

the vernier dial on the HP 33120A Signal Generator. This experiment proceeded in a manner similar to experiment one, and comparable results were obtained with 21 trials.

The best case achieved was in trial 8, with parameters: IGain=300, IIGain –300 and SG=1.75, at a reduction in amplitude of 19.02 Hz. This reduction was not at node 26, as was used in experiment 1. For this mode the controller was unable to reduce the amplitude of node 26's y-directed motion. Good results were obtained from node 41, the diagonal counterpart of node 26 at the other extreme of the truss. This trial was the best overall case during the course of research and testing. Graphical results from this trial are presented in the figures below.

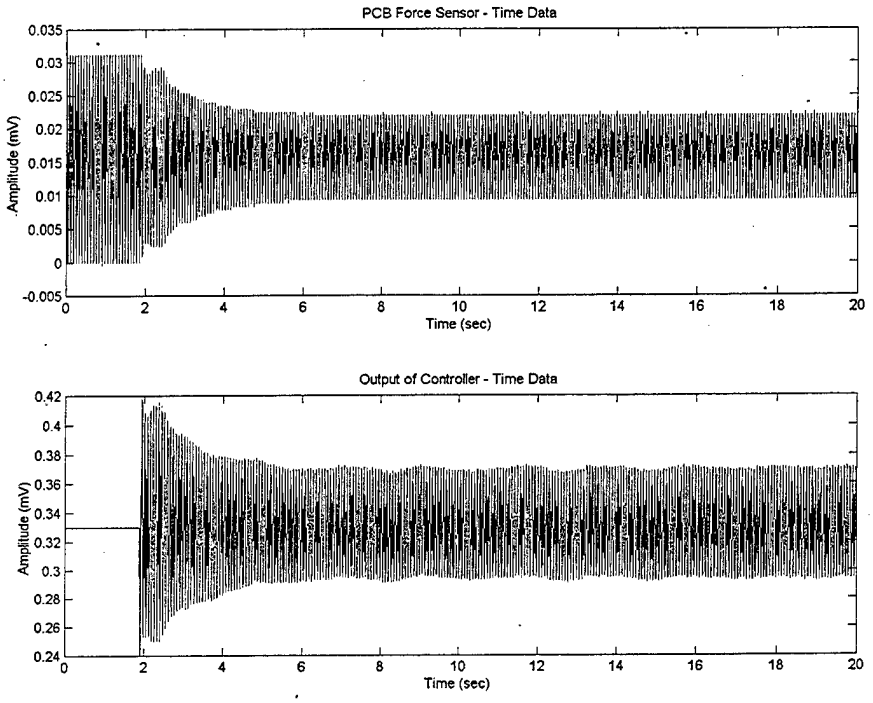


Figure 31. Exp. 2, Trial 8 Controller Response

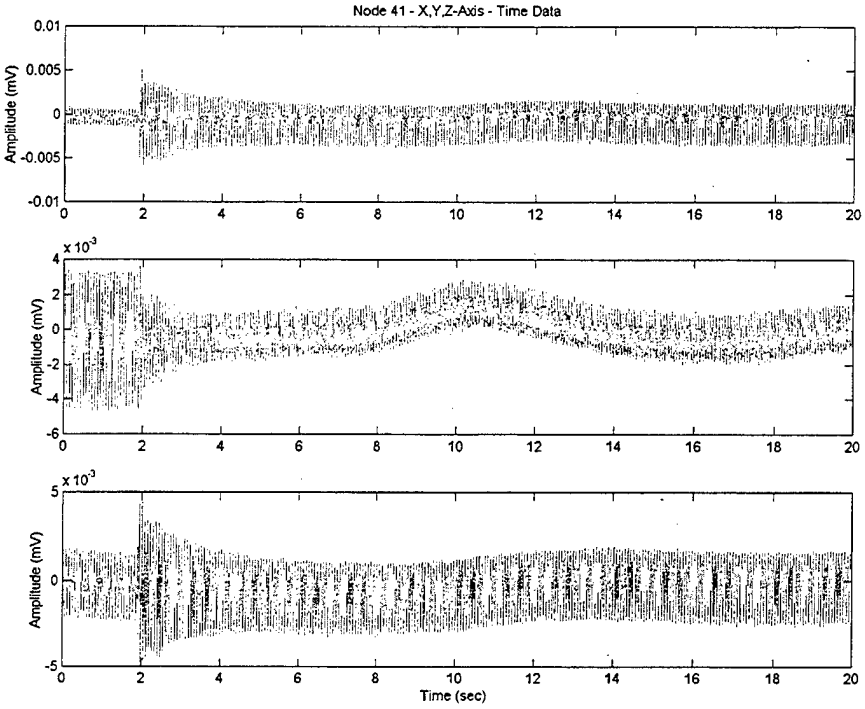


Figure 32. Exp. 2 Trial 8 Node 41 Response

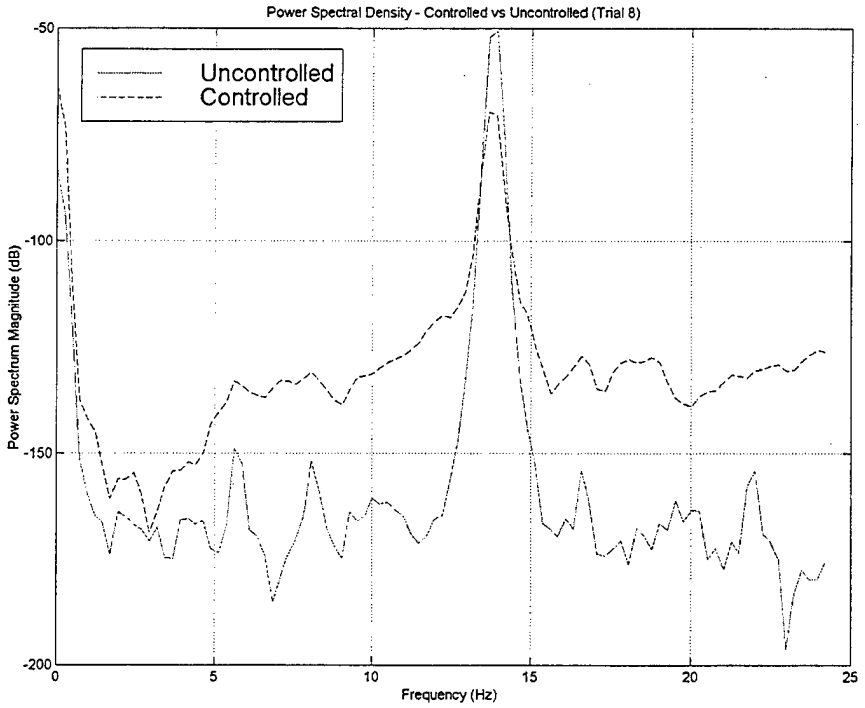


Figure 33. Exp. 2 Trial 9 Node 26 Response and PSD

4. Experiment 3, Repeat of Vlattas And Johnson, 16.75 Hz

The third experiment was a repetition of the test performed by LT Scott Johnson and LT John Vlattas [Ref. 14]. This experiment searched for a target frequency that had a 90-degree phase offset from the actuator to the sensor. In their controller, band pass filters were used (high pass filters are currently used) to target a specific frequency. In this case 16.75 Hz was found to have a 90-degree offset. Use of the DSA to examine this region with greater frequency resolution confirmed that this was the location of the 90-degree offset.

With the exception of the filters and the new dSPACE system, the experimental setup was identical. However, when starting the system, it was found that a 100 Vpp signal, as reported in Reference 14, did not produce a noticeable signal from the force transducer. Once the amplitude was raised to 1.0 Vpp a measurable signal appeared. Further increasing the amplitude of the generated disturbance signal to 1.2 Vpp allowed the use of the laser diode in qualitative study.

Sixteen trials were conducted in this experiment. The best results that were obtained were during trial 8, with gain parameters of IGain=300, IIGain=-100, SG=1.75 giving a reduction of 17.93 dB. These results were consistent with the approximately 15 dB reduction achieved previously [Ref. 14, p. 98]. A different value of IIGain was found to have the best response during this experiment than the runs performed by Lieutenants Johnson and Vlattas. This is possibly due to a slight shift in the frequency response owing to the moving of the truss from the Newport vibration isolation table during the salvage efforts after the flood in early 1999.

5. Experiment 4, Two Modes With Variable Phasing

In this experiment, the first two fundamental natural frequencies, 12.50 and 13.81 Hz were targeted simultaneously to observe the broadband performance of the controller and to verify that the superposition principle is in effect. According to the superposition principle, linear signals may be separated, have linear operations performed on them and then be recombined to produce a single output.

To create the dual frequencies, the dSPACE controller, created in Simulink, was modified to include a dual-frequency signal generator, with the capability of varying the starting phase of one of the signals. This phase variation was intended to explore the superposition idea just discussed. The ControlDesk experimental layout was also modified to allow quasi real-time adjustment of these parameters. The modification to the Simulink model is shown in Figure 34.

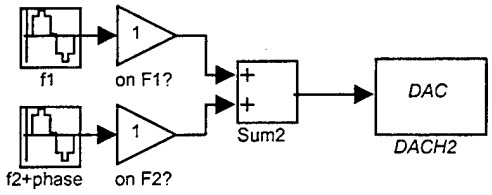


Figure 34. Dual-Frequency Modification to ST_Controller

For this experiment, twelve trials were performed, at six different phasings. In each case, an uncontrolled run was performed, and a controlled run, operating with the parameters: IGain=300. IIGain=100 and SG=1.75. The phase offset for the second frequency was varied from zero to 240-degrees in 60-degree increments. The results from the six cases were all similar, and resulted in a reduction in amplitude of the highest peak of 7.29 to 9.26 dB. This reduction was in the first frequency, 12.50 Hz. The second

frequency, 13.81 Hz showed a reduction of about 15 dB. Each case showed similar results, verifying the superposition principle as applied here. The beating phenomenon from two close sinusoidal signals, and their spectral response are shown in Figure 35 and Figure 36.

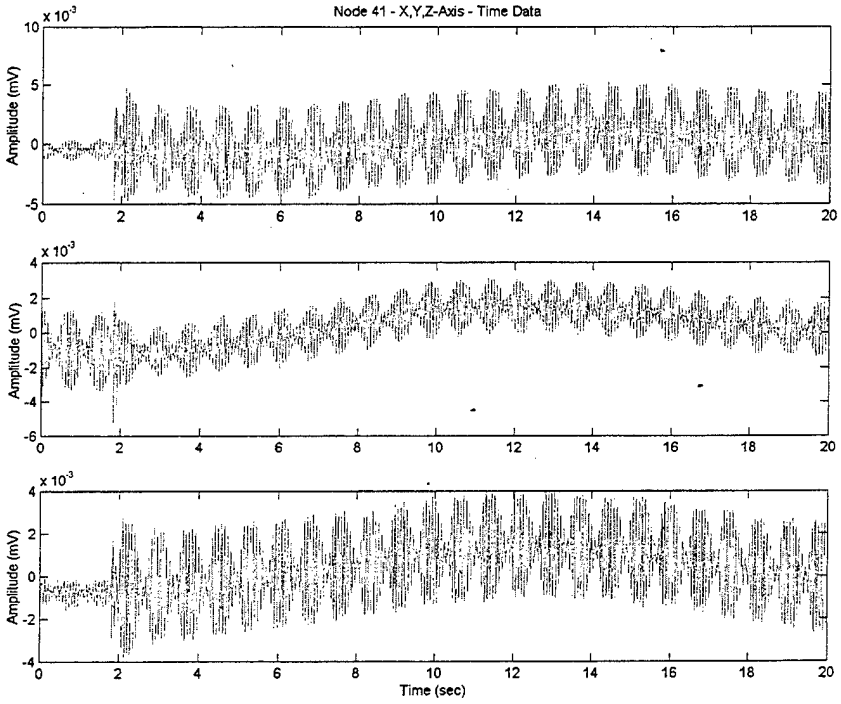


Figure 35. Exp. 4 Trial 51 Node 41 Response

An entire representative case of the dual-frequency experiment is shown in Appendix O, Figure 57 through Figure 59.

6. Experiment 5, Random Noise

The final experiment performed was intended to further assess the broadband performance of the truss controller with random noise. The truss was excited with white

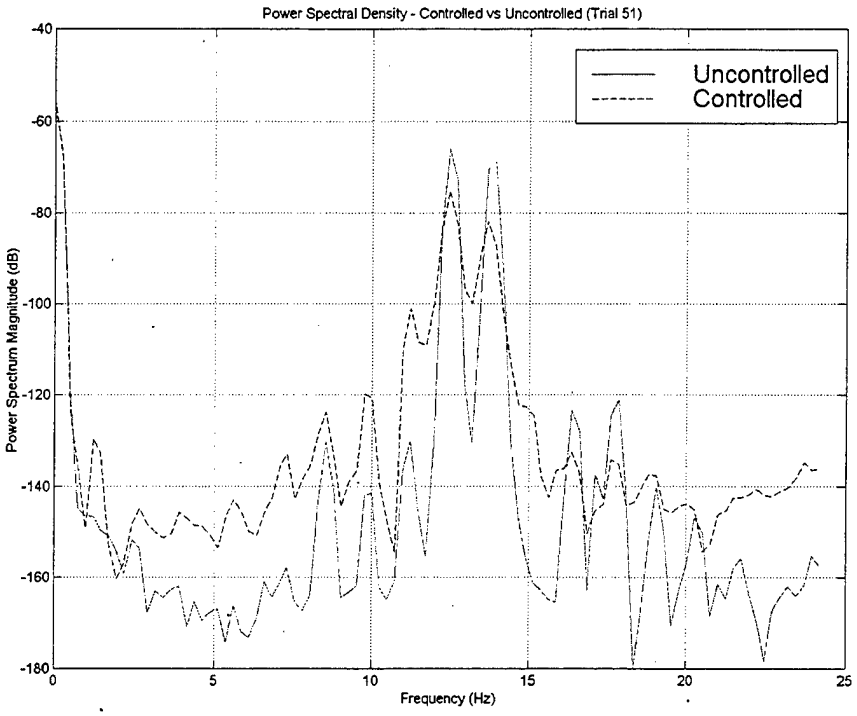


Figure 36. Exp. 4 Trial 52 Power Spectral Density

noise, generated from the HP 35665A Dynamic Signal Analyzer. The DSA white noise function has a band-pass filter inherent in the output that makes the majority of the power in a desired spectral region. The region selected from this experiment was 10-35 Hz, as this contains a large number of modes. Outside of this band, the noise input tapers off linearly with respect to the logarithm of the magnitude [Ref. 28, p. 4-213]. The random signal generated from the DSA was echoed to the oscilloscope. Examination of this signal revealed that it was not truly random, but consisted of a series of sinusoidal fragments of randomly varying frequency.

Twenty trials were performed. The first ten trials were performed uncontrolled, the second set of ten trials were performed with the controller active, with the parameters: IGain=300, IIGain=100 and SG=1.75. The sets of runs were processed with the data

processing routines and their PSD's averaged to find a mean value of the controlled and uncontrolled case. Examination of the data revealed a variation of +/-5dB.

The change of the highest peak of the uncontrolled cases to the highest of the controlled cases was 6.69 dB. This number is not truly representative of the broadband performance of the system. The PSD is shown in Figure 37. It shows that the response is significant in the targeted range, and reductions of 10-25 dB were observed in the frequencies targeted for the first three experiments. There is an increase of about 10dB in the natural frequency that was added by the installation of the LPACT (about 8 Hz). This shows that the addition of the LPACT, especially the integrated spring, is significant to the frequency response of the truss, and should be examined in further experiments.

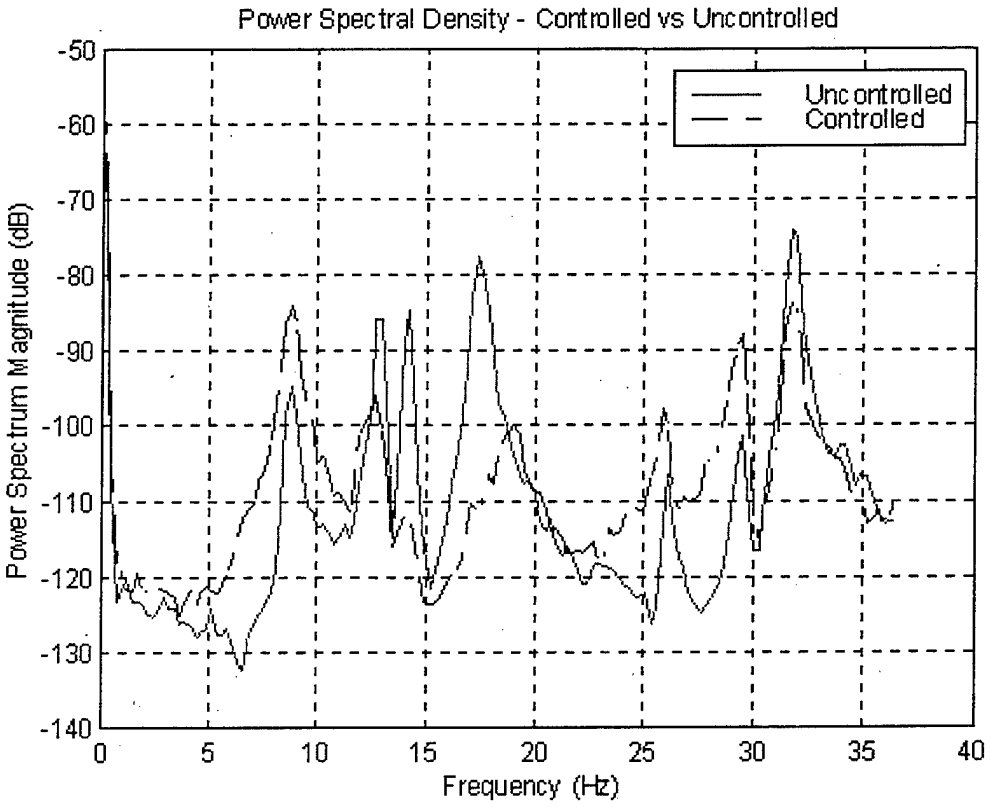


Figure 37. Exp. 5 Random Noise Average Power Spectral Density

D. ANSYS FEM

1. Model Setup and Initiation

As discussed in Chapter IV, the SOLID5 elements are not permitted to have a modal analysis performed on them if they use PIEZ data tables. Also, if a modal analysis were performed on a model before the inclusion of the PIEZ tables, an error in the database would ensue that would prevent the running of subsequent transient analyses. To work around this issue, a copy of the active NPS space truss active model was used to delete the PIEZ data tables and obtain the modified truss natural frequencies. This model was not used for any transient analysis. Future versions of ANSYS will permit the PIEZ data tables to be used in modal analysis [Ref. 27]. It is not known if the transient problem will be resolved.

The natural frequencies obtained by this analysis were significantly different from the original modal frequencies. This is in part due to the added mass of the LPACT and control apparatus (The mass is approximately 20-percent above the bare truss mass). The natural frequencies obtained from the modified FEM are presented in Table 8.

Mode	ANSYS Frequency	Actual Frequency
1	9.187	13.113
2	11.299	17.125
3	11.743	29.688
4	17.781	72.075
5	26.332	89.371
6	33.010	96.153
7	59.797	103.046
8	77.324	110.338
9	77.830	135.820
10	81.508	140.448

Table 8. Natural Frequencies of Truss With Active Components

Upon examining the mode shapes from the postprocessor, it was determined that the first natural frequency was located at the LPACT strut and was due to the 2.5 kg mass vibrating there. This mode was not present in this form in the real truss as the LPACT has a stiffer adapter to the truss than the relatively long and flimsy aluminum beams in the FEM. However, in the real truss, the LPACT has a spring that does have a natural frequency of about 8-10 Hz (as seen in Appendix A). This frequency was seen in the frequency determination with the DSA and subsequent random vibration experiment, previously in this chapter.

The second mode, at 11.299 Hz is a combination of the effects of the LPACT addition and the original first mode. Examination of the mode shapes revealed that the second mode combined the properties of the LPACT with the first original mode. The third modified mode shape was observed to be a combination of the first and second bare truss mode shapes. This produces a diagonal motion of the truss, and is consistent with the experimental results qualitatively observed with the laser mounted above node 52.

When comparing the ANSYS modified natural frequencies with those experimentally obtained from the work of LT Johnson and LT Vlattas [Ref. 14, p. 110] it can be seen that there is a large discrepancy in the frequencies. It is believed that the previous experiment did not target the natural frequency that was added by the LPACT, and that the two analytical modes at 11.30 and 11.75 Hz were molded together in the experimental analysis.

As stated in Chapter IV, the SOLID5 element is connected to the structure with the use of rigid regions that constrain the element from contracting as it expands in the axial direction. This will induce an artificial stiffness into the model if not examined.

The NPS space truss FEM was modified to delete all of the components not directly connected to the active strut, leaving only the SOLID5 elements and a few BEAM4s. A static analysis was performed, setting a 100V signal on the end of the piezoelectric actuator.

As reported in previous work [Ref. 14, p. 25-28] and shown in Figure 7, the actuator has an expansion of 50μm for an applied voltage of 100V. Under the static analysis, the actuator had a displacement of only 37.1-μm, 74-percent of the expected value. The inverse of the 74-percent results shows that an increase of 35-percent is required to match the experimental data, roughly the value of Poisson's ratio. The value of d_{33} in the PIEZ data table was increased to 9568 from 7100 to account for this effect. This patch was only possible because of the prior work that measured the exact operating properties of the piezoelectric actuator.

The actual truss active element was preloaded with shims to give a compressive bias for the force transducer and piezoelectric actuator. In the ANSYS model, this effect was not included due to the mathematical modeling of the SOLID5 elements. A limiter was included, but was set to +/- 60V.

The runs were designed to run for 4.0 seconds of simulated time. As the time-step interval of $1/20^{th}$ the driving frequency was used, a limit was discovered that affected the final time used. ANSYS will only permit the performance of 1000 transient steps in an analysis. This required the reduction of the total time for the 16.75 Hz data set. If longer data runs are required, it would be necessary to either save the final conditions and apply them at the start of a new data run or use a larger value for the time step.

The active control APDL macro, "Act_Truss.inp," was modified to allow the expanded use of APDL's capabilities. The macro was divided into two programs, one that set up the initial parameters and saved the calculated data, and the other to perform the transient analysis. In this manner the workstation was directed to perform 10-40 data runs at a time (limited by the time desired for completion and analysis by the author). As each data run was approximately 15 minutes and 40 seconds, this allowed use of the computer processor in the off-hours.

The files saved by the overarching APDL macro were ASCII files that contained the node 26 y-direction motion, sensor input and controller output. A MATLAB algorithm was developed to process the data in a rapid manner. This algorithm is included as Appendix P. PROC_ANSYS.M.

2. ANSYS Series 1: 11.30 Hz

In this series of data runs, the second natural frequency, 11.299 Hz was targeted. As discussed above, this frequency is the combination of LPACT effects and the first original natural frequency. The overarching APDL macros were directed to examine variations in IGain, IIGain, and finally to attempt to get a better performance by backing slightly off the IGain limit and attempting to find a best operating condition. Detailed electrical schematics for the sensor and actuator were not available, so trial and error to determine the polarity of the control loop was performed. It was observed that negative values for SG provided control, indicating a polarity correction in the sensor of actuator.

The ANSYS model displays different instability characteristics than those of the NPS space truss. The voltage limiter drives the control signal into saturation, which

causes transients to be seen as disturbances on the amplitude graphs and as side lobes on the PSD chart. Depending on whether the model was limited by IGain or IIGain, different results were observed. The IIGain term causes larger side lobes on the order of 80dB. Examples of these effects are shown in Appendix Q. ANSYS RESULTS, Figure 61 and Figure 62. The IIGain term causes a different settling time constant that the IGain dominated cases. An example of a slightly saturated data run is presented as Figure 63.

In all, 27 data runs were performed for the first series. The best overall results were obtained with trial 16, with a reduction in node 26 amplitude of 22.4521 dB. This value was the best obtained for the five different series of data runs. This data run is presented as Figure 38 and Figure 39. Trial 16 was run during the examination of the double integral effects alone, and represents a position only style of feedback.

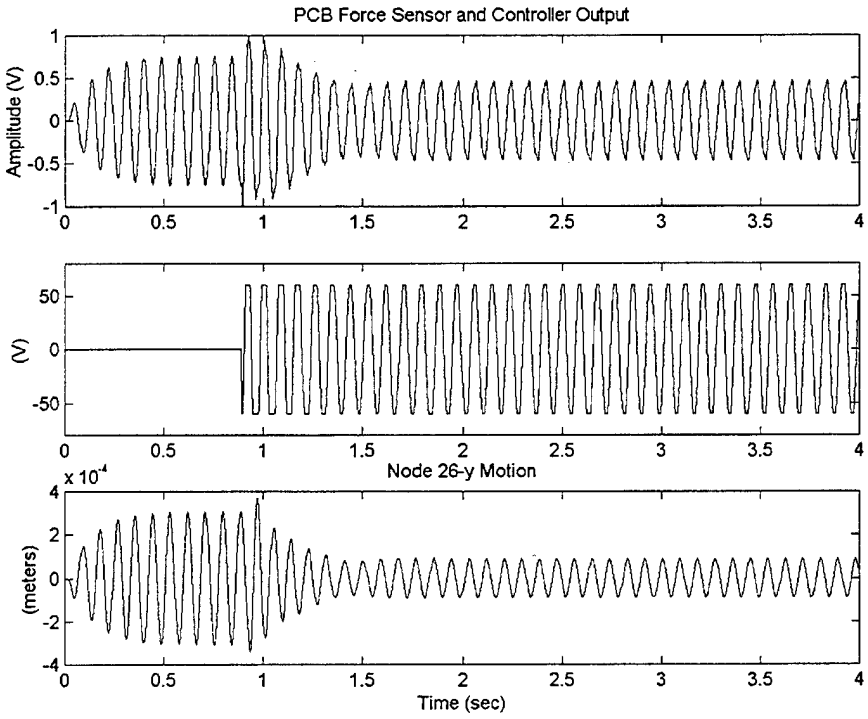


Figure 38. Series 1 Trial 16 Results

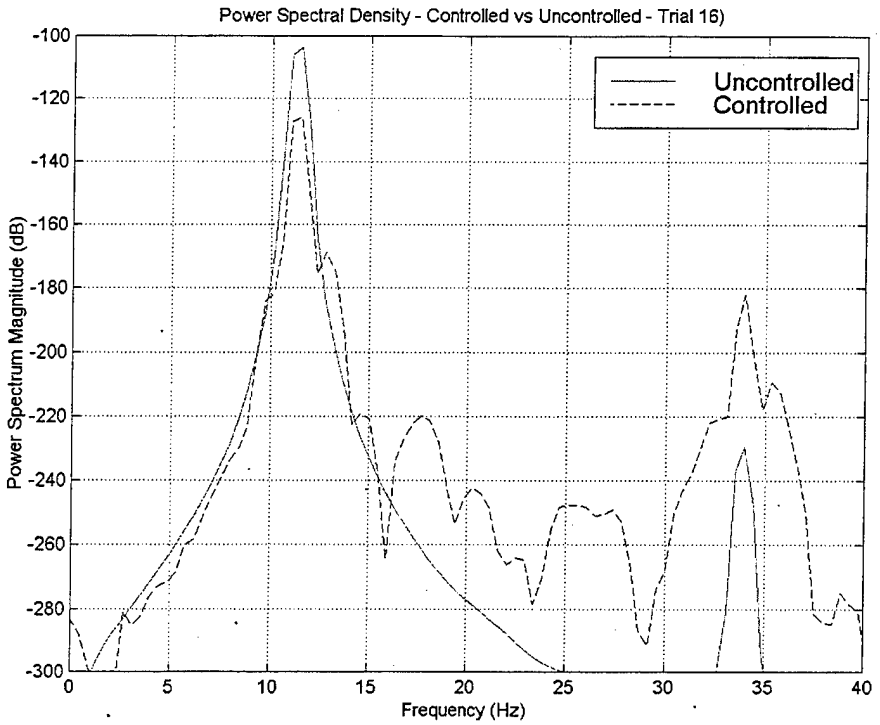


Figure 39. Series 1 Trial 16 Power Spectral Density

From the above figure, an increase can be observed at the 33-Hz frequency range, at the location of the sixth analytical mode. As can be observed, there is an increase outside the range of second and third natural frequencies, the ones targeted for specific analysis. This effect may be due to the Rayleigh damping term used that is effective in the desired range, and gives poor broad frequency range damping characteristics.

3. ANSYS Series 2: 11.75 Hz

The second series targeted 11.75, the third natural frequency. Due to the decaying nature of the LPACT signal as frequency increases, smaller amplitudes were observed for the uncontrolled case in this series, as compared to the first. Consequently, a slightly larger gain was applied to the controller to reach saturation. At this frequency, the model displayed similar saturation transient characteristics as the series performed at 11.30 Hz.

The APDL macro ran the same series of gain constants as the first series. Additional runs were performed at the end to find a limit with both gain terms. Of the 34 runs performed, the best results obtained were in trial 34, showing a reduction of 18.0605 dB. This trial was at the saturation limit. An unusually low value was observed for trial 22, reflecting that the gain constants were in error. The APDL Macro has already been modified when this was discovered, but the consistency in the nearby trials supports this conclusion. Care is required when programming the APDL macro. Future studies could examine the use of easy to read data tables to minimize errors.

4. ANSYS Series 3: 16.75 Hz

Series three was intended to observe the experiment performed by LT Johnson and LT Vlattas [Ref. 14]. Initial runs on this series revealed that very different results would be obtained. As the negative gain was applied, an increase in the amplitude was observed. An example of this was trial 3, shown in Appendix Q as Figure 65. Further, the controller drove itself into saturation at gain levels that were far below those of the first two series (the first series became saturated at –150 and –125 values of SG, series three was saturated at 50). As the now positive SG was raised, the model entered a saturation region and did not re emerge to go to the other side, as was the case in the previous two series.

Adjusting the three gain settings did little to correct this occurrence. The limit between stable and unstable was difficult to find. The best results obtained were in trial 4, showing a reduction of 19.6464 dB at the target frequency. This run, however, caused the amplitude of the signal in the 8-12 Hz band to increase by 100 dB, a significant effect.

Trial 4 is shown as Figure 66. Trial 5, the saturation and instability limit, is shown in Figure 67.

5. ANSYS Series 4: Dual Frequencies

The dual frequency experiment was repeated in this series of data runs. The 11.30 Hz and 11.75 Hz frequencies were used. As these two frequencies are close in value, a long time beating phenomena was expected. For this reason, and due to the good results obtained with series one and two, the sampling time was raised to $1/10^{th}$ the highest frequency, with a sample run of 8.0 seconds to allow the beating phenomena to be observed.

These trials were run with gain settings that were satisfactory for both frequencies to prevent instability. As the only noise in a digital system is due to round off error, these results were much more consistent than the parallel experiment performed on the actual NPS space truss. For all trials, results of about 15 dB were obtained. A typical controlled trial is presented in the appendix as Figure 68.

6. ANSYS Series 5: Random Noise

For the final data series, twenty runs were performed, ten uncontrolled and ten controlled. The sampling frequency was set at 200 Hz, at 4.8 seconds to maximize the collection within the 1000 time step limit imposed by ANSYS. ANSYS has a random number generator function in APDL that was used to obtain the random signal.

The PSDs obtained were averaged, and their values compared. The reduction from the peak uncontrolled to the peak controlled value was 6.0371 dB. The PSD is

presented as Figure 40. It can be observed that the originally targeted frequencies, in the 11-13 Hz band show reductions of about 15 dB, which is consistent with series four.

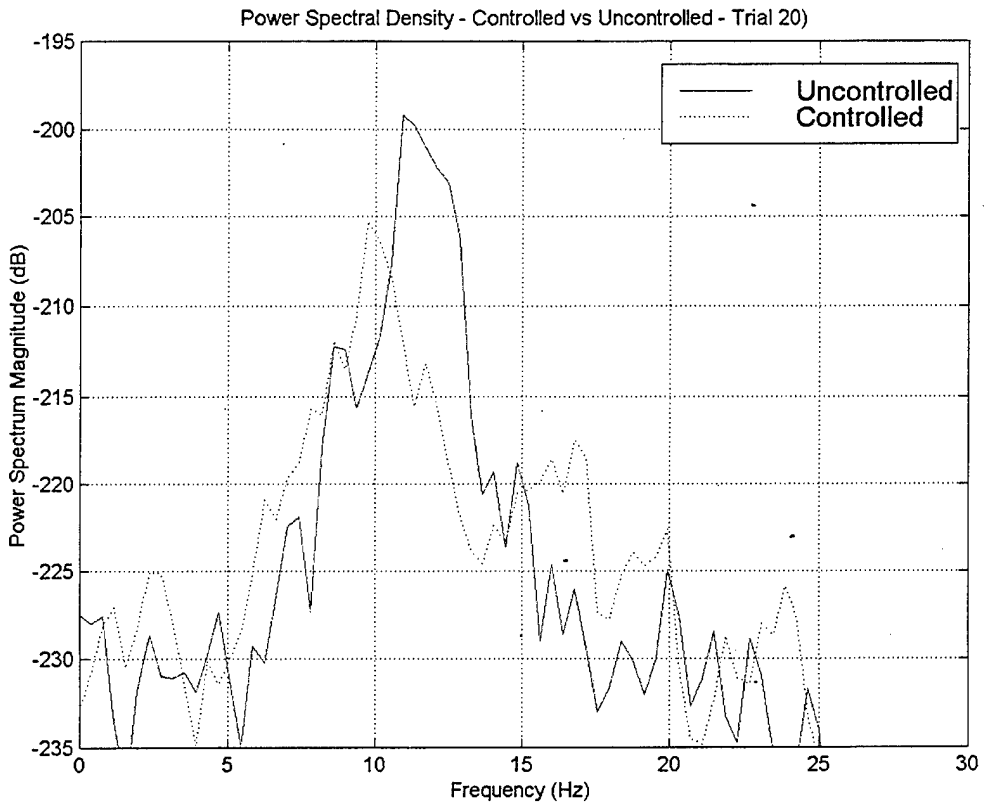


Figure 40. Series 5: Random Power Spectral Density

The purely random noise in this series did not produce as large a reduction that the first two series due the filtering action. The digital filters required three steps of consistent data to get a good data trend. The purely random nature of the disturbance signal did not give ample time for the filters to perform properly. Both the ANSYS results and those obtained from the actual truss show an increase in the 8-10Hz range owing to the LPACT contribution.

E. COMPARISON AND SUMMARY OF RESULTS

The best cases from each run, experimental and analytical are presented here, as Table 9.

Experimental	Analytical	IGain	IIGain	SG	Results (dB)
Exp. 1 Trial 15	--	300	-400	1.75	18.5410
--	Series 1 run 16	0	5,000	-150	22.4521
Exp. 2 Trial 8	--	300	-300	1.75	19.0202
--	Series 2 run 34	300	40,000	-100	18.0605
Exp. 3 Trial 8	--	300	-100	1.75	17.9299
--	Series 3 run 4	300	0	25	19.6464
Exp. 4 Trial 51	--	300	100	1.75	9.2564
--	Series 4 run 31	300	0	-75	15.5659
Exp. 5	--	300	100	1.75	6.6880
--	Series 5	300	0	-75	6.0371

Table 9. Summary of Results

The results obtained with the analytical model in ANSYS behaved in a manner similar to those obtained by experiment. Overall, the amplitudes in the analytical model were slightly higher than those in the experiment, by about 2-3 dB. The system gain term, SG, was inclusive of the 20-to-1 Trek 50/750 voltage amplifier that was present in the actual space truss. This reflects most of the difference in the values of SG when comparing the analytical results with the experiment. When this factored in, the analytical model has the ability increase SG by a factor of three before saturation.

When the tests were driven into instability, a difference was discovered between the ANSYS model and the actual truss. The actual truss would become very unstable, and oscillate at the saturation limit for a period that was much longer than it was driven. The analytical model would simply enter saturation, and change states at each period, at the same frequency as the driving signal. Both showed high frequency transient responses at the saturation limits.

The actual truss and the ANSYS model exhibited an overall reduction in the frequency range of interest, and an increase in the magnitude at other frequencies, especially at the LPACT frequency range (8-10 Hz). Overall, the ANSYS model, while more stable owing to its saturation vice instability showed poorer response in the higher frequency bands. A more detailed model, including better damping effects, may better capture this effect.

Finally, the analytical results were more consistent overall than the experimental results. This was due to the computational nature of the experiment, and could not include the random effects that any real system would experience (i.e. noise).

VI. CONCLUSIONS AND RECOMMENDATIONS

By using IDIFF control for the Naval Postgraduate School Space Truss, vibration reductions of 15-20 dB were realized with the use of the dSPACE data acquisition and processing system. The ANSYS actively controlled finite element model showed similar, qualitative results, amplitude reductions of 18-22 dB. This demonstrated that the method devised to actively control an FEM is valid, and can predict the control authority in a structure with active components installed.

The ANSYS model was devoid of system noise, except that which occurs with small round off errors that are present in computer processors. This may be the cause of the slightly better results and different stability and saturation characteristics observed in the experiments and data runs.

The analytical method developed using ANSYS, which is commercially available, has room for refinement. The modeling of the actuator with a SOLID5 element using rigid regions to attach it to the structure caused some inconsistencies in the initial data test runs. A calculational patch was used to remedy this problem, but further analysis is required to refine the method for integrating an active control device into a structure's finite element model.

The responsiveness of the NPS space truss to the IDIFF controller was positive (i.e. reduced the amplitude) in the examined degrees of freedom. Other DOFs showed large increases in amplitude. The addition of more actuators to the NPS space truss is recommended to allow better vibration reduction in the truss as a whole. Other controllers could be explored that would lead to larger reductions. An option for a

prospective controller is to use the MATLAB finite element model contained herein to create a dynamic observer using the accelerometer input to create a full state feedback controller for the truss.

A feedback-style controller is only effective to a limit (as the sensor signal lowers it reduces the controller output) and is not usually able to fully eliminate the vibration in a structure if continuously driven. A feed-forward controller, such as the clear box system developed for the Ultra-Quiet Platform, another active vibration control experiment in the SRDC, may be implemented as a means to accomplish this. These controllers rely on some knowledge of the disturbance, either by sensing directly or by estimation, and are limited in that regard.

The purely random nature of the signals created in the ANSYS data series 5 did not allow a good look at the spectrum, as the random nature prevented the filters from being able to have the proper effect. In order to demonstrate a good capability at all frequencies, a frequency sweep algorithm should be written with APDL to perform an analysis over the range of the frequencies desired. This is a task best suited for APDL. As each run was about 15 minutes in length, an entire data set would take days to perform.

The method created within of using ANSYS to perform an active control simulation on a structural FEM can be used on any manner of structure. It is able of being added to any pre-existing ANSYS model, or created with a new model. In this manner large vibration-susceptible structures, such as the International Space Station can be examined with active vibration reducing devices to determine their effect on the structure when in orbit.

LIST OF REFERENCES

1. Interview between Robert L. Boucher, Senior Engineer, Boeing Information, Space and Defense Systems, Huntington Beach, CA, and the author, 15 March, 1999

2. dSPACE GmbH, *DS1103 PPC Controller Board Installation and Configuration Guide*, dSPACE GmbH, Germany, March 1999

3. *ANSYS Theory Reference*, Section 11.1, ANSYS 5.5 Online Help, SAS, IP, Inc.

4. The Math Works, *MATLAB Reference Guide*, The Math Works, October 1992

5. McClelland, Robert, Lim, Tae W. ,Bosse, Albert B. and Fisher, Shalom, "Implementation of Local Feedback Controllers for Vibration Suppression of a Truss Using Active Struts", *Proceedings of the SPIE 1996 Symposium on Smart Structures and Materials*, pp. 452-461, February 1996

6. *NRL Space Truss Assembly Schematics*, January 1997

7. NRL Code 8200 Interoffice Memo from Bosse, Albert to Shatzer, Blaine ,"NPS Truss Assembly," 13 March 1997

8. *Newport Vibration Control System Instruction Manual*, Newport Corporation, Irvine, CA 1991

9. Holland, Richard, *Design of Resonant Piezoelectric Devices*, page 10, Massachusetts Institute of Technology Press, 1969

10. Halliday, David, Resnik, Robert and Walker,Jearl, *Fundamentals of Physics Extended*, Fifth Edition, p. 796, John Wiley & Sons, Inc, 1997

11. Bronowicki, Allen J., *Smart Structures, Work in Progress Report*, p. 3, TRW Space and Electronics Group, Spacecraft Technology Division, 14 September, 1994

12. *Piezo Performance Test Procedures and Documents (Technical Note TN 59E/2),* Physik Instrumente, Waldbronn, Germany, 12 February 1997

13. Project 95-C-5003 Task 4, Report Number 3614-004-2, *Finite Element Modeling of Smart Structures*, p. 40, MRJ, Inc. 1996

14. Johnson, Scott E. and Vlattas, John, *Modal Analysis and Active Vibration Control of the Naval Postgraduate School Space Truss*, Master's Thesis, Naval Postgraduate School, Monterey, California, June 1998

15. Kwon, Young W. Bang, Hyochoong, *The Finite Element Method Using MATLAB*, CRC Press, 1997

16. Blakely, Ken, *MSC/NASTRAN Basic Dynamic Analysis*, p18, MacNeal-Schwendler Corporation 1993

17. PCB Piezotronics, "PCB Piezotronics Products – PFS Division 208B02," [http://www.pcb.com/products/pfs/pfs208b02.html]

18. *LPACT and Electronics User Manual*, Planning Systems Incorporated, Melbourne Controls Group, Melbourne, FL, 1997

19. Andberg, Brent K., *"Modal Testing and Analysis of the NPS Space Truss,"* Master's Thesis, Naval Postgraduate School, Monterey, California, September, 1997

20. dSPACE GmbH, *Real-Time Interface (RTI and RTI-MP) Implementation Guide*, dSPACE GmbH, Germany, March 1999

21. dSPACE GmbH, *ControlDesk Experiment Guide*, dSPACE GmbH, Germany, March 1999

22. dSPACE GmbH, "Variable Description File", *dSPACE Help Desk (On-Line Help)*, From ControlDesk Version 1.0, dSPACE GmbH, Germany, 1999

25. SAS IP, Inc., *ANSYS Commands Reference*, Tenth Edition, Available as On-Line Help, SAS IP, Inc.

24. SAS IP, Inc., *ANSYS Elements Reference*, Tenth Edition, Section 4.5, Available as On-Line Help, SAS IP, Inc.

25. Barzanti, R., EC2400 Lecture Notes, p. 97, Naval Postgraduate School, Fall 1997

26. SAS IP, Inc., *ANSYS Structural Analysis Guide*, Fourth Edition, Section 5.12, Available as On-Line Help, SAS IP, Inc.

27. E-mail correspondence received from Sheldon Imaoka, Collaborative Solutions, Inc., 23 September, 1999

28. E-mail correspondence received from Sheldon Imaoka, Collaborative Solutions, Inc., 04 October, 1999

29. Hewlett-Packard, *HP35665A Dynamic Signal Analyzer Operator's Reference,* Chapter 4, September, 1993

30. dSPACE GmbH, *MLIB/MTRACE MATLAB-dSPACE Interface and Trace Libraries*, dSPACE GmbH, Germany, March 1999

31. Preumont, Andre, Dufour, Jean-Paul and Malekian, Christian, "Active Damping by a Local Force Feedback with Piezoelectric Actuators," *Journal of Guidance, Control and Dynamics*, Volume 15, No. 2 March-April 1992

32. Physik Instrumente, "P842, P-843 Preloaded Open & Closed Loop LVPZT Translators," [http://www.physikinstrumente.com/pztactuators/1_16t.html]

33. PCB Model 208B02 S/N 15021 Calibration Data Sheet, PCB Piezotronics, 5 Mar 1998

APPENDIX A. NPS SPACE TRUSS CHARACTERISTICS AND COMPONENTS

Mass Properties of the Bare and Modified Truss:

Part Name	No. In Bare Truss	No. In Mod Truss	Component Masses (kg)	Mass Bare Truss (kg)	Mass Mod. Truss (kg)
Node Balls	52	52	0.0663	3.445	3.445
Longerons	100	100	0.0448	4.475	4.475
Diagonals	61	58	0.0522	3.181	3.025
LPAC Strut	0	1	2.2760	0.000	2.276
Act. Strut #1	0	1	0.2900	0.000	0.290
Screw	322	322	0.0019	0.607	0.607
Total Mass Truss				11.708	14.118

Table 10. NPS Space Truss Mass Properties [After Ref. 14]

Bare Truss Natural Frequencies

Mode Number	Modal Testing [Ref. 14]	MATLAB FEM	ANSYS
1	14.64	14.13	14.25
2	16.26	15.44	15.57
3	30.41	28.72	28.93
4	33.97	32.04	32.26
5	62.93	60.23	60.76
6	74.54	72.24	72.93
7	80.66	79.71	81.67
8	101.01	97.41	96.61
9	126.23	120.21	115.41
10	135.97	129.68	122.26

Table 11. NPS Space Truss Bare Natural Frequencies

Property	Aluminum	Steel
Outer Radius	3.968 mm	3.975 mm
Inner Radius	3.078 mm	Solid
Inertia	1.242e-10 m^4	1.957e-10 m^4
Cross-sectional Area	1.96856e-5 m^2	4.96e-5 m^2

Table 12. Truss Element Properties

COMPONENTS

Piezoelectric Translation Model P-848-30 S/N

Signal [Volts]	Expansion [Microns]	Contraction [Microns]	Hysteresis [Microns]	Hysteresis [Percent]
0.00	0.00	0.00	0.00	0.00
1.00	3.48	6.43	2.94	5.98
2.00	7.58	12.55	4.97	10.12
3.00	12.16	18.38	6.21	12.64
4.00	17.20	23.86	6.66	13.56
5.00	22.53	29.06	6.53	13.28
6.00	27.96	33.90	5.95	12.10
7.00	33.44	38.38	4.94	10.05
8.00	38.83	42.43	3.61	7.34
9.00	44.05	46.07	2.02	4.11
10.00	49.14	49.14	0.00	0.00

Table 13. Expansion and Contraction Data for Model P-843.30 [From Ref. 14]

Open Loop Travel (0-100V)	45 µm +/- 20%
Closed Loop Travel	45 µm
Stiffness	33 N/µm +/- 20%
Force Generation (Blocked)	1500 N +/-20%
Push/Pull Force Capability	800/300 N
Torque Limit (at tip)	350 mNm
Capacitance	5.4µF +/- 20%
Dynamic Operating Current Coefficient	15 µA/(Hz-µm)
Unloaded Resonant Frequency	10 kHz +/- 20%
Operating Temperature Range	-20 to +80 °C
Mass (w/o cables)	53 g
Length	73 mm

Table 14. P-843.30 Operating Characteristics [Ref. 32]

Planning Systems Incorporated LPACT [Ref. 18] S/N CML-030-020-1

Item	Value
Force Constant (K_f)	5.5 lb/amp
Max. Current	1 amp
Coil Resistance	9 ohms
Flexure Natural Frequency (ω_n)	8 to 10 Hz
Flexure Modal Damping(ξ)	~3 % (or critical) without force loop, up to >100% with force loop on
Stroke	±0.2 inches
Stroke at 10 Hz for 3 lbs. output force	0.1 inches
Gravity Offset Spring Rate	2.4 lb/in
Allowable Strut Diameter	1.000 ± 0.01"
LPACT Envelope	3.8" OD x 4.86" height (including strut clamp and accelerometers)
LPACT Total Weight	4.0 lb.
LPACT Proof Mass Weight	2.9 LB
LPACT Model (low frequency) (refer to Figure 2 for measured FRF from current to force of LPACT)	$\frac{Output\,Force(lb)}{Current\ \text{Command}(amp)} = \frac{K_f s^2}{s^2 + 2\xi\omega_n s + \omega_n}$
Servo Amp Model	$\frac{\text{Current (amp)}}{\text{Servo Command (V)}} = K_a = 0.1\ \frac{\text{amp}}{\text{V}}$
Force Loop Model (see section 3.3 for definition of terms)	$\frac{\text{Servo Amp Voltage Command (volts)}}{\text{Pr oof Mass Accel (g)}} = \frac{K_{pre}\ K_{rt} K_{force} s^2}{(s + w_{pre})(s + w_{rt})^2}$
Rate Loop Model (see section 3.3 for definition of terms)	$\frac{Servo\ \text{Amp Voltage Command (volts)}}{\text{Primary Accel}(g)} = \frac{K_{pre} K_{rt} K_{rate} w_{rate} S_r\ s^2}{(s + w_{pre})(s + w_{rt})^2 (s + w_{rate})}$

Table 15. LPACT Characteristics

Cable Assembly cable	connect to LPACT Electronics (all on rear panel)	connect to LPACT Component
Black Coax	'To Coil' (banana plug to BNC adapter)	6" Blue Pigtail from coil (BNC)
Blue Coax marked with Red Tape	'From Secondary Accelerometer' (BNC)	Secondary Accelerometer on Proof Mass (microdot)
Blue Coax	'From Primary Accelerometer' (BNC)	Primary Accelerometer on Co-Locate Ring (microdot)

Table 16. LPACT Electronics Connectivity Guidelines [From Ref. 18]

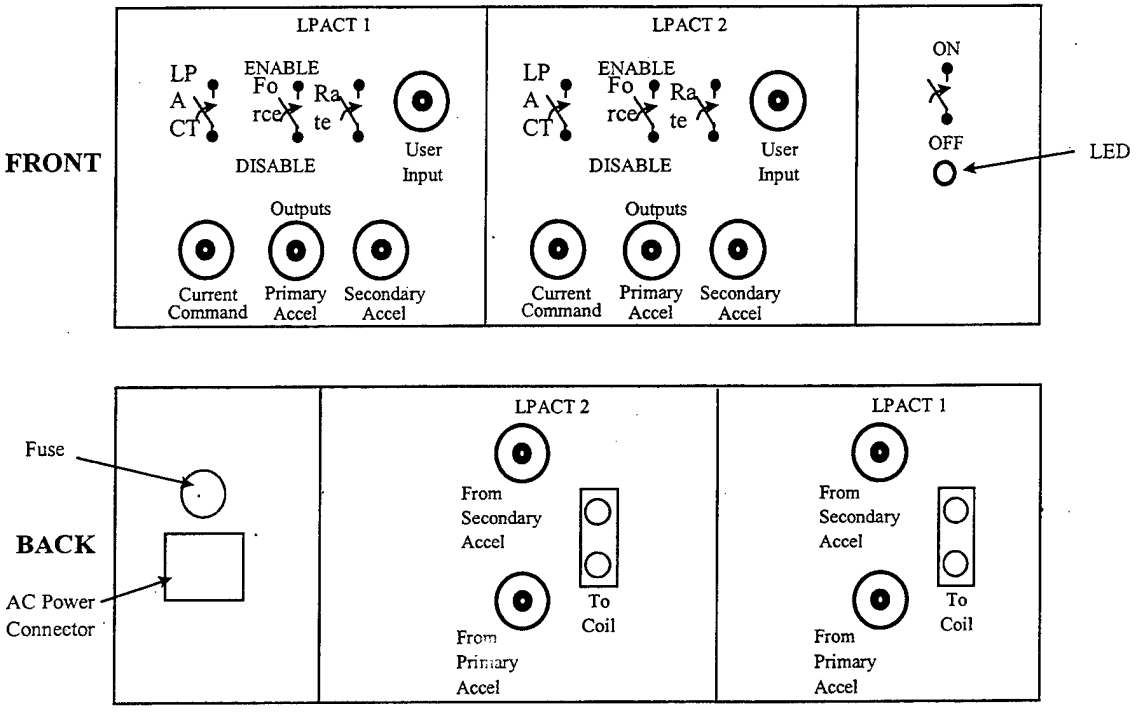


Figure 41. LPACT Control Electronics Rear Panel [From Ref. 18]

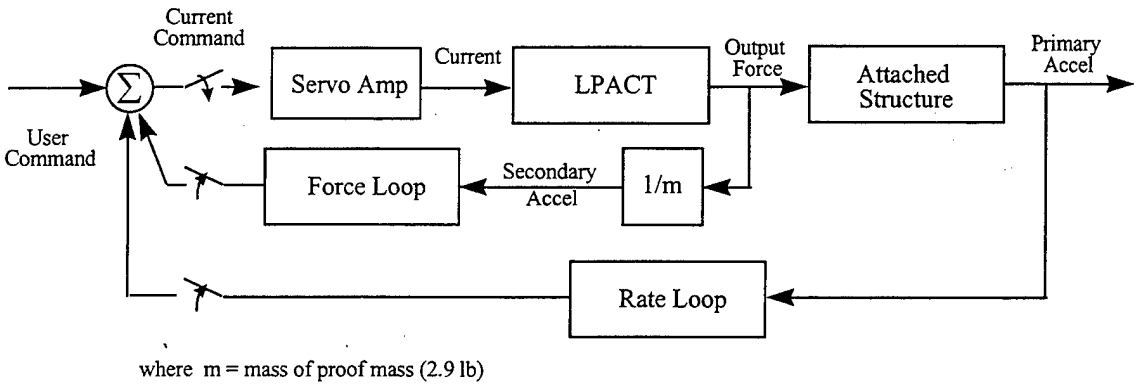


Figure 42. LPACT System Level Block Diagram [From Ref. 18]

PCB Piezotronics Model 208B02 General Purpose ICP Force Sensor S/N 15021

Sensitivity (Specification)	50 mV/lb (11240 mV/kN)
Sensitivity (Measured)	50.80 mV/lb [Ref. 33]
Dynamic Range	-10 lb to 100-lb
Stiffness	1.0 kN/μm
Temperature Range	-54 to 121 °C
Sensing Element	Quartz

Table 17. PCB Model 208B02 Operating Characteristics [Ref. 17]

PCB® Piezotronics Type 484B Signal Conditioner S/N 2086

Notes
Unity Gain
Set CPLG to DC & Bias to 6 V

Kistler Instrument Corp. Accelerometers:
(Note: g = 9.807 m/s^2)

Type	Serial Number	+ x-axis	+ y-axis	+ z-axis	
8690C10	C112398	495	490	494	mV/g
8690C10	C112399	487	490	490	mV/g
8690C10	C112400	499	500	494	mV/g
8690C10	C112401	497	491	505	mV/g

Kistler Instrument Corp. Signal Conditioners (Multi-Channel Couplers):

Type	Serial Number
5124A (twelve channel)	C74930

Trek Voltage Amplifier:

Type	Serial Number	Notes
Trek 50/750	none	Required calibration on 10 September 1998. Mini-cal to verify setting and linearity Performed by author on 2 July, 1999 Two channels that can be used with both active struts.

dSPACE GmbH, Germany, S/N 3192

CPU	Motorola PowerPC running at 300 Hz
Memory	128MB RAM
Input Channels	20 ADC
Ouput Channels	8 DAC 0-10V

Hewlett Packard HP 33120A Signal Generator S/N
settings

Hewlett Packard HP 54601A Digital Oscilloscope S/N 3134A02713
Four Channels
Frequency response up to 100 MHz

Hewlett Packard HP 35665A Dynamic Signal Analyzer S/N
hookups and settings

SOFTWARE DOCUMENTATION

Program	Version
MATLAB	5.2.1430
dSPACE RTI1103	3.3
dSPACE MLIB/MTRACE	3.1.1
dSPACE ControlDesk	1.0
ANSYS/Multiphysics	5.5.2

Table 18. Software Documentation

APPENDIX B. FINAL_ROOT_NPS.M

```
% Final_root.m modified from
% Example 8.9.5
% Modified by LT Carey M. Pantling for ME4613 Final project
% eigenanalysis of NPS Space Truss
% Bulids systems mass and stiffness matrices
%
% Variable descriptions
%   coord = global x,y and z coordiates of each node
%   nd = nodal connection vector
%   k = element stiffness matrix
%   kk = system stiffness matrix
%   m = element mass matrix
%   mm = system mass matrix
%   ff = system force vector
%   index = a vector containing system dofs associated with each element
%   bcdof = a vector containing dofs associated with boundary conditions
%   bcval = a vector containing b c values associated with
%           the dofs in 'bcdof'
%--------------------------------------------------------------------
%
clear;
nel=161;                % number of elements
nnel=2;                 % number of nodes per element
ndof=6;                 % number of dofs per node
nnode=52;               % total number of nodes in system
sdof=nnode*ndof;        % total system dofs

load nps_coord.dat        %  52 by 3, 52 nodes x,y,z
load nps_node.dat             %  161 by 2, 161 elements, node1 node2
load nps_bcdof.dat          %  24 by 2, 24 degrees constrained

coord=nps_coord;              nd=nps_node;
bcdof=nps_bcdof(:,1);         bcval=nps_bcdof(:,2);

el=70E9;            % GPa    elastic modulus
area=1.9686982E-5; % m^2    cross-sectional area
xi=1.242217E-10;    % m^4    moment of inertia of cross-section
rho=2800;           % kg/m^3 mass density per volume

ff=zeros(sdof,1);          % initialization of system force vector
kk=zeros(sdof,sdof);       % initialization of system matrix
mm=zeros(sdof,sdof);       % initialization of system matrix
index=zeros(nel*ndof,1);   % initialization of index vector

for iel=1:nel     % loop for the total number of elements
   index=feeldof(nd(iel,:),nnel,ndof);  % extract system dofs
   % compute element stiffness and mass matrices
   [k,m]=feframe3(el,xi,area,rho,coord(nd(iel,1),:),coord(nd(iel,2),:));
   kk=feasmbl1(kk,k,index); % assemble each element matrix into system
      mm=feasmbl1(mm,m,index);
end
```

```
% add the concentrated masses at the nodal translation dof
for iel=1:nnode
   start=(iel-1)*ndof+1;
   fini=start+2;
   % add lumped masses for the node points
   mm(start:fini,start:fini)=mm(start:fini,start:fini)+(0.0663)*eye(3);
   %kg added to each node as concentrated mass
end

% apply boundary conditions
[kk,mm,ff]=feaplyc2(kk,mm,ff,bcdof,bcval);

[V,D]=eig(kk,mm);
fsol=diag(D);
fsol=sqrt(fsol)/2/pi;
% purge NaN's
fsol(isnan(fsol))=[]
```

APPENDIX C. FEFRAME3.M

```
function [k,m]=feframe3(el,xi,area,rho,xyz1,xyz2)

%     From feframe2.m by Young Kwon
%     Modified for 3D by LT Carey M. Pantling
%     Purpose:
%     Stiffness and mass matrices for the 3-d frame element
%     nodal dof {u_1 v_1 w_1 theta_1 theta_2 theta_3
%                u_2 v_2 w_2 theta_4 theta_5 theat_6}
%     element stiffness per FEM/MATLAB by Kwon, etc. p 264
%     adds lumped masses at ends for the screws and fittings
%
%  Synopsis:
%     [k,m]=feframe3(el,xi,leng,area,rho,xyz1,xyz2)
%
%  Variable Description:
%     k - element stiffness matrix (global)
%     m - element mass matrix (global)
%     el - elastic modulus
%     xi - second moment of inertia of cross-section
%     area - area of beam cross-section
%     rho - mass density (mass per unit volume)
%     xyz1 - coordinates of first node (1x3)
%     xyz2 - coordinates of second node (1x3)
%     kl - element stiffness matrix (size of 12x12) (local)
%     ml - element mass matrix (size of 12x12)
%     dx,dy,dz - differences in the three axes
%     leng - element length
%     ul,vl,wl - local unit vectors (global vectors i,j,k)
%     OA - off axis vector for local unit vector definition
%     a,b,c - constants for the stiffness matrix
%     J=rotational inertia term =2*xi
%     G=shear modulus = E/2.6 assumes nu=0.3

% compute elemetal dimensional data
dx=xyz2(1)-xyz1(1); dy=xyz2(2)-xyz1(2); dz=xyz2(3)-xyz1(3);
leng=(dx^2+dy^2+dz^2)^0.5;

% compute element rotation matrix
%      local aligned with x along axis, y,z orthogonal
%      since tube segments, y,z do not matter
% DCM defined as {x}g=g[c]l*{x}l; g[c]l is the transformation matrix
ul=[(dx/leng),(dy/leng),(dz/leng)];
cr=cross(ul,[0,0,1]);
mag=(cr(1)^2+cr(2)^2+cr(3)^2)^0.5;
if mag~=0 % if {ul}x{i}~=0 then
   OA=[0,1,0;-1,0,0;0,0,1]*ul'; % rotate z-90degrees
   else
      OA=[0,0,-1;0,1,0;1,0,0]*ul'; % rotate y-90 degrees
end
vl=cross(ul,OA); vl=vl/(vl(1)^2+vl(2)^2+vl(3)^2)^0.5;
wl=cross(ul,vl); wl=wl/(wl(1)^2+wl(2)^2+wl(3)^2)^0.5;
```

```
DCM=[ ul(1) , vl(1) , wl(1);...
      ul(2) , vl(2) , wl(2);...
      ul(3) , vl(3) , wl(3) ];
r=zeros(12);
r(1:3,1:3)=DCM;
r(4:6,4:6)=DCM;
r(7:9,7:9)=DCM;
r(10:12,10:12)=DCM;
T=r';

% stiffness matrix at the local axis
kl=zeros(12);
G=el/2.6;
J=xi*2;
a1=el*area/leng;     % axial compression
a2=J*G/leng;         % axial torsion
b1=12*el*xi/leng^3; % pure bending due to deflection-y
b2=6*el*xi/leng^2;  % pure bending due to deflection-y and rotation z
b3=2*el*xi/leng;     % bending due to rotation z
c1=b1; % tube therefore symmetric
c2=b2; % tube
c3=b3; % tube
kl(1:6,1:6)=[a1,  0,  0,  0,  0,  0;...
             0, b1,  0,  0,  0, b2;...
             0,  0, c1,  0,-c2,  0;...
             0,  0,  0, a2,  0,  0;...
             0,  0,-c2,  0,2*c3, 0;...
             0, b2,  0,  0,  0,2*b3];
kl(1:6,7:12)=[-a1,  0,  0,  0,  0,  0;...
               0,-b1,  0,  0,  0, b2;...
               0,  0,-c1,  0,-c2,  0;...
               0,  0,  0,-a2,  0,  0;...
               0,  0, c2,  0, c3,  0;...
               0,-b2,  0,  0,  0, b3];
kl(7:12,1:6)=kl(1:6,7:12)';
kl(7:12,7:12)=[a1,  0,  0,  0,  0,  0;...
               0, b1,  0,  0,  0,-b2;...
               0,  0, c1,  0, c2,  0;...
               0,  0,  0, a2,  0,  0;...
               0,  0, c2,  0,2*c3, 0;...
               0,-b2,  0,  0,  0,2*b3];
% stiffness matrix at the global axis
k=T'*kl*T;

% compound mass matrix per MSC/Nastran
ml=zeros(12,12);
mass=rho*area*leng/12;
ml(1:3,1:3)=5*eye(3); ml(7:9,7:9)=ml(1:3,1:3);
ml(1:3,7:9)=eye(3);   ml(7:9,1:3)=ml(1:3,7:9);
ml=mass*ml;
% add mass for the fittings, with no change in stiffness
ml(1:3,1:3)=ml(1:3,1:3)+0.015*eye(3);
ml(7:9,7:9)=ml(7:9,7:9)+0.015*eye(3);
% mass in the global system
m=T'*ml*T;
```

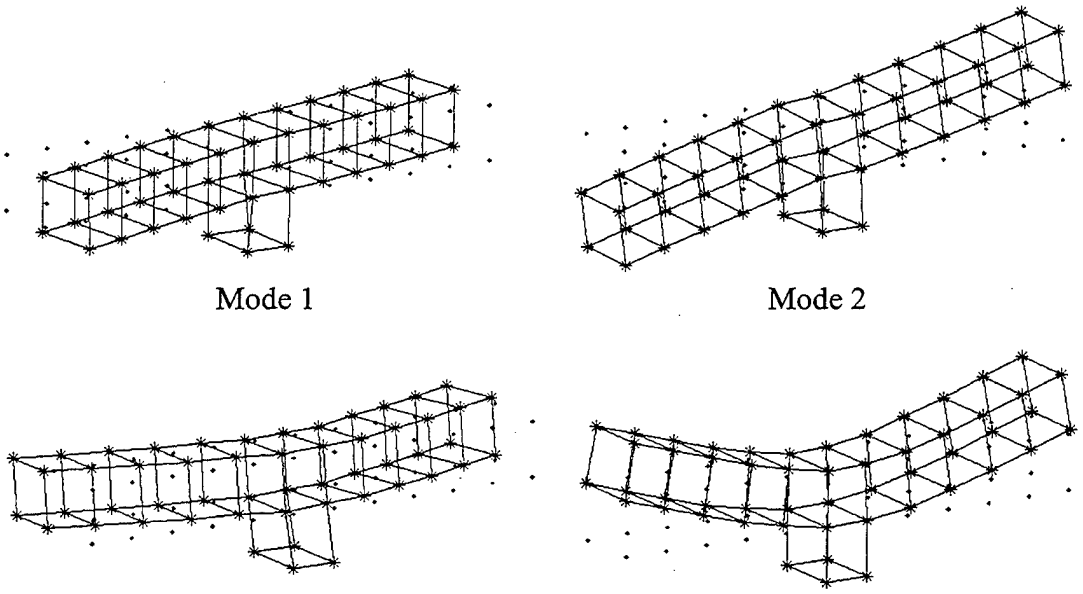
APPENDIX D. NPS_MODES.M

```
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% NPS_MODES.M Plot code for mode shape plotting %
% By LT Carey M. Pantling                       %
% Uses Final_root_nps.m for its base program    %
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% plot the modes
figure(2);
amp=2.0;
mode=V(:,259);
disp=nps_coord;
for index=1:nnode
   start=(index-1)*6+1;
   disp(index,1)=disp(index,1)+amp*mode(start);   % x-coord
   disp(index,2)=disp(index,2)+amp*mode(start+1); % y-
   disp(index,3)=disp(index,3)+amp*mode(start+2); % z-
end

% isolate the real components of the eigenvectors
disp=real(disp);

% form the longeron combinations
plot3(disp(1:2,1),disp(1:2,3),disp(1:2,2),'b*-');
hold on
plot3(nps_coord(1:2,1),nps_coord(1:2,3),nps_coord(1:2,2),'r.');
view(-30,20);
axis([-2 3 -1 2 -1 2]);
plot3(disp(3:14,1),disp(3:14,3),disp(3:14,2),'b*-');
plot3(nps_coord(3:14,1),nps_coord(3:14,3),nps_coord(3:14,2),'r.');
view(-30,20);
axis([-2 3 -1 2 -1 2]);
plot3(disp(15:26,1),disp(15:26,3),disp(15:26,2),'b*-');
plot3(nps_coord(15:26,1),nps_coord(15:26,3),nps_coord(15:26,2),'r.');
view(-30,20);
axis([-2 3 -1 2 -1 2]);
plot3(disp(27:28,1),disp(27:28,3),disp(27:28,2),'b*-');
plot3(nps_coord(27:28,1),nps_coord(27:28,3),nps_coord(27:28,2),'r.');
view(-30,20);
axis([-2 3 -1 2 -1 2]);
plot3(disp(29:40,1),disp(29:40,3),disp(29:40,2),'b*-');
plot3(nps_coord(29:40,1),nps_coord(29:40,3),nps_coord(29:40,2),'r.');
view(-30,20);
axis([-2 3 -1 2 -1 2]);
plot3(disp(41:52,1),disp(41:52,3),disp(41:52,2),'b*-');
plot3(nps_coord(41:52,1),nps_coord(41:52,3),nps_coord(41:52,2),'r.');
view(-30,20);
axis([-2 3 -1 2 -1 2]);

% cross ties
for index=1:26
   line(1,:)=disp(index,:);
```

```
   line(2,:)=disp(index+26,:);
   plot3(line(:,1),line(:,3),line(:,2),'b*-');
end

% verticals
for index= [1 2 27 28]
   line(1,:)=disp(index,:);
   line(2,:)=disp(index+7,:);
   plot3(line(:,1),line(:,3),line(:,2),'b*-');
end
for index= 3:14
   line(1,:)=disp(index,:);
   line(2,:)=disp(index+12,:);
   plot3(line(:,1),line(:,3),line(:,2),'b*-');
end
for index= 29:40
   line(1,:)=disp(index,:);
   line(2,:)=disp(index+12,:);
   plot3(line(:,1),line(:,3),line(:,2),'b*-');
end
hold off
title('Mode 4');
print -djpeg mode4ml.jpg
```

APPENDIX E. NPS SPACE TRUSS MODE SHAPES

The first four mode shapes obtained with MATLAB are presented first. The deformed shapes are shown with the undeformed node locations for comparison. The first four mode shapes generated by ANSYS are presented second, with both the deformed and undeformed shapes.

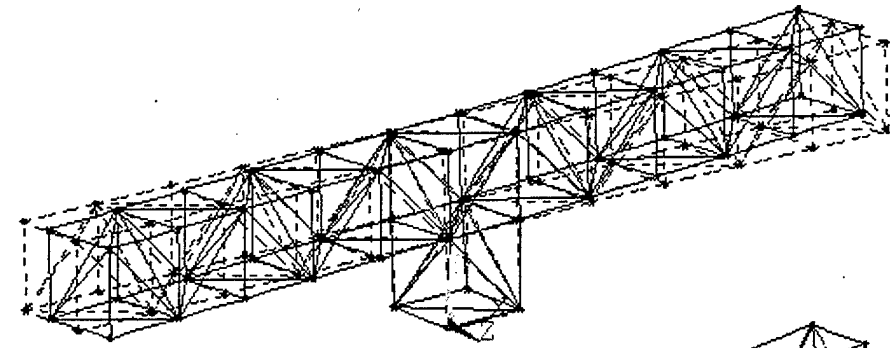


Mode 1 Mode 2

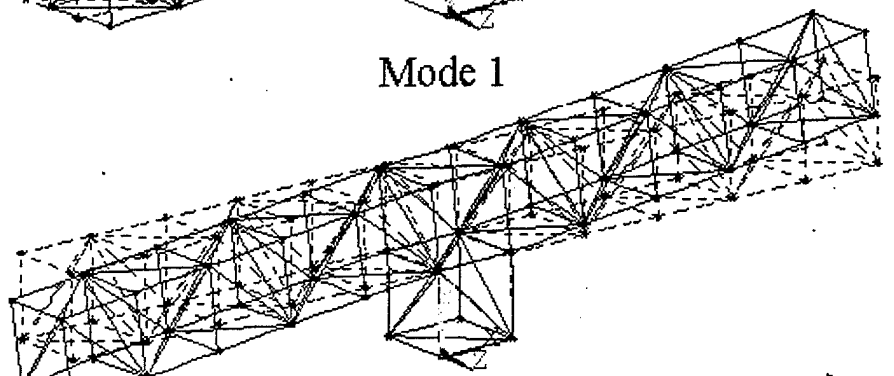
Mode 3 Mode 4

Figure 43. NPS Space Truss Mode Shapes with MATLAB

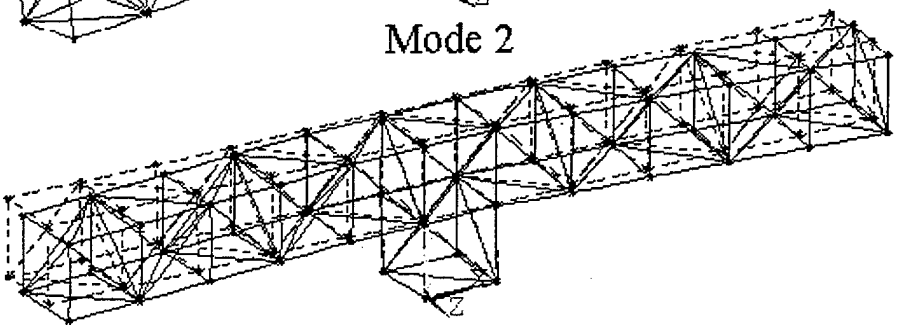
The mode shapes that were generated from ANSYS were directed to show both the deformed and undeformed states. The undeformed states appear as dotted lines in the following figures.



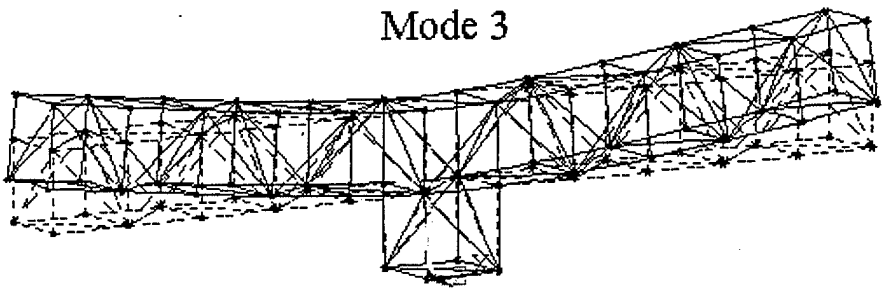
Mode 1



Mode 2



Mode 3



Mode 4

Figure 44. NPS Space Truss Mode Shapes with ANSYS

APPENDIX F. NPS_STRAIN.M

```
% Strain energy determination, by LT Carey M. Pantling
% Uses Final_root_nps for the determination of the system
% Variables required in workspace
%     kk    - global stiffness matrix
%     ff    - global force vector
%     mm    - global mass matrix, not used except as part of program
%     bcdof - DOF for boundary conditions
%     bcval - values for the BCs at DOF above
%     coord - matrix (nnode by 3) of nodal coordinates
%     nd    - matrix (nel by 2) of nodal connections
%     el    - Elastic Modulus
%     area  - cross sectional area
% Program Variables
%     disp  - displacements from solution
%     delsq - elemental change in length, squared

% get the strain energies for the mode eigenvectors
disp=coord;  % reset displaced positions
amp=1.0;
mode=V2(:,260);
% get the new nodal positions
for index=1:nnode
   start=(index-1)*6+1;
   disp(index,1)=disp(index,1)+amp*mode(start);    % x-coord
   disp(index,2)=disp(index,2)+amp*mode(start+1); % y-
   disp(index,3)=disp(index,3)+amp*mode(start+2); % z-
end
% get the elemental strain energies U=0.5EAd^2/L
for index=1:nel
   index;
   % original element length
   dx=coord(nd(index,2),1)-coord(nd(index,1),1);
      dy=coord(nd(index,2),2)-coord(nd(index,1),2);
      dz=coord(nd(index,2),3)-coord(nd(index,1),3);
   leng=(dx^2+dy^2+dz^2)^0.5; % original length
   % new length
   delx=disp(nd(index,2),1)-disp(nd(index,1),1);
   dely=disp(nd(index,2),2)-disp(nd(index,1),2);
   delz=disp(nd(index,2),3)-disp(nd(index,1),3);
   newleng=(delx^2+dely^2+delz^2)^0.5; % new length
   change=newleng-leng;
   str_eng(index)=0.5*el*area/leng*change^2;
end
[str,in]=sort(str_eng);
% flip the matrix
for index=1:nel
   strout(index)=str(162-index);
   inout(index)=in(162-index);
end
real(strout(1:30))'
inout(1:30)'
```

APPENDIX G. TRUSS.INP

```
! ******************************************************
! * Truss Control APDL Program for ANSYS version 5.5
! * For simple sinusoidal disturbance
! * Written by LT Carey M. Pantling
! * With assistance by Sheldon Imaoka
! * Last Modified 16 July 1999
! ******************************************************

! First Load Truss model with mesh and BC's

! ******************************************************
! * Define Variables
! ******************************************************

*set,FREQ,16.75                    ! disturbance frequency
*set,CS,0.0                        ! control signal to piezo, initial zero
*set,PV,node(0.300,-0.004,0.004)   ! Piezo voltage node for application
*set,RS,                           ! read signal from piezo, defined only
*set,SV,node(0.150,-0.004,0.004)   ! Sensor voltage node for detection
*set,G1,300                        ! Control gain constants, see below
*set,G2,100
*set,G3,0.0
*set,OLDRS,0.0                     ! the old RS
*set,OLDINT,0.0                    ! previous integral, IC 0.0
*set,OLDFINT,0.0
*set,OLDDBL,0.0                    ! prev DI, IC 0.0
*set,STRTTIME,0.0                  ! Start time = 0.0 seconds
*set,FINITIME,1.0                  ! Finish Time in seconds
TIMESTEP=1/FREQ/20                 ! TIMESTEP
*set,PI,acos(-1)
*set,DISPNODE,node(0.375,0.0,0.0)! DISPNODE = the node where 41 is in truss
*set,CSCheck                       ! tracks the control signal
*set,RSCheck                       ! tracks the read signal
finish

! ******************************************************
! * Set up load disturbance, at least 20 steps per period
! ******************************************************
*set,NUMSTEP,nint(FINITIME/TIMESTEP)
*set,FORCFUN,
*set,FFP,
*dim,FORCFUN,array,NUMSTEP    ! array for forcing function
```

```
*dim,FFP,table,NUMSTEP          ! table for plotting
*do,ICOUNT,1,NUMSTEP
 FORCFUN(ICOUNT)=sin(2*PI*FREQ*((ICOUNT-1)*TIMESTEP))
 FFP(ICOUNT)=sin(2*PI*FREQ*((ICOUNT-1)*TIMESTEP))
*enddo
*set,NODE41Y
! creates an array to get the node 41-y displacements
*dim,NODE41Y,table,NUMSTEP
*dim,CSCheck,table,NUMSTEP      ! gets the control signal
*dim,RSCheck,table,NUMSTEP      ! copies the sensed signal
*dim,INTcheck,table,NUMSTEP
*dim,DBLcheck,table,NUMSTEP
*dim,FINTchk,table,NUMSTEP

! ****************************************************
! * Do loop for loading at each time step
! ****************************************************
*do,ICOUNT,1,NUMSTEP
! Set current time (actually, end of current step)
 *set,CURRTIME,STRTTIME+ICOUNT*TIMESTEP

 /solu                          ! enter the solution processor
 *if,ICOUNT,eq,1,then
  antype,trans,new              ! Start new transient analysis
 *else
  antype,trans,rest             ! Restart or continue transient analysis
 *endif
 time,CURRTIME                  ! set current time for ANSYS solution
 deltim,TIMESTEP                ! set delta-t
 f,DISPNODE,fx,FORCFUN(ICOUNT)  ! apply force to DISPNODE node
 d,PV,volt,CS                   ! apply CS voltage to PV node
 allsel,all                     ! select everything to prepare for solve
 solve
 finish
 *if,ICOUNT,gt,100,then         ! turns on control at 100th step
  G3=-11.0
 *endif

 /post1                         ! enter post processor
 *get,RS,node,SV,volt           ! get the value of the sensor voltage
 RSCheck(ICOUNT)=RS             ! saves current RS in table
 *get,NODE41Y(ICOUNT),node,DISPNODE,u,x ! get the tip displacement
 finish

! ************************************************
```

```
 ! * Control Law implementation, for next time step
 ! ************************************************
 *set,INT,OLDINT+(RS+OLDRS)/2*TIMESTEP  ! approx the integral
 *set,FINT,0.95*OLDFINT+INT-OLDINT      ! digital high pass filter
 *set,DBLINT,OLDDBL+(FINT+OLDFINT)/2*TIMESTEP
 INTcheck(ICOUNT)=INT
 FINTchk(ICOUNT)=FINT
 DBLcheck(ICOUNT)=DBLINT
 OLDRS=RS
 OLDINT=INT
 OLDFINT=FINT
 OLDDBL=DBLINT
 CS=(G1*FINT+G2*DBLINT)*G3          !CS=RS*(G1/s+g2/s^2)*G3
 CSCheck(ICOUNT)=CS
*enddo                              !ends the loop, ready for the next time step

! Plot the results to screen
/erase
*vplot,,NODE41Y(1)

! ************************************************
! * Write "NODE41Y" results to text file "truss.out"
! ************************************************
!*cfopen,piezo5_tip_c132,out,
!*vwrite,NODE41Y(1),
!(E10.3)
!*cfclos
```

APPENDIX H. ACTIVE ELEMENT INTEGRATION DETAILS

This appendix contains excepts from the log file used in the creation of the NPS space truss active model. The commands have been commented to provide clarity. All commands that had no relevance on the model creation were removed (for example, display commands)

```
/PREP7
! create element for SOLID5
ET,3,SOLID5
! set parameters
KEYOPT,3,1,0
KEYOPT,3,2,0
KEYOPT,3,3,0
KEYOPT,3,5,0
! unmesh line and delete
LCLEAR,   101
GPLOT
LDELE,   101, , ,1
! Define Steel material
UIMP,4,EX, , ,200000000000,
UIMP,4,DENS, , ,7920,
UIMP,4,NUXY, , ,.3,
UIMP,4,MURX, , ,1,
UIMP,4,PERX, , ,1,
! place the WP at node 8
KWPAVE,    8
! rotate to strut axis
wpro,-45.000000,,
wpro,,,45.000000
wpro,,,45.000000
! create keypoints for the elements
K, ,0.037545,,,
K, ,0.189145,,,
K, ,0.199145,,,
K, ,0.2324,,,
K, ,.2754,,,
KWPAVE,    2
K, ,-.037545,,,
! create lines to connect keypoints
LSTR,    8,    53
LSTR,    53,    54
LSTR,    55,    56
LSTR,    57,    58
LSTR,    58,    2
KWPAVE,    54
! shift WP for 3-axis alignment
wpro,,,45.000000
wpro,,,45.000000
! create the local coord system
CSWPLA,11,10
! create the volume for the sensor
BLC4,-.005,-.005,.01,.01,.01
WPSTYLE,,,,,,,,0
KWPAVE,    56
! create the volume for piezo
BLC4,-0.005,-.005,.01,.01,.043
! defines real constant steel beams
R,8,.496e-4,.195e-9,.195e-9,.004,.004, ,
RMORE, , , , , , ,
! select new lines mesh as 1 element
! alumiumum lines
FLST,5,2,4,ORDE,2
FITEM,5,101
FITEM,5,165
CM,_Y,LINE
LSEL, , , ,P51X
!*
CM,_Y1,LINE
CMSEL,,_Y
LESIZE,_Y1, , ,1,1,
CMDEL,_Y
CMDEL,_Y1
! steel lines
FLST,5,2,4,ORDE,2
FITEM,5,162
FITEM,5,164
CM,_Y,LINE
LSEL, , , ,P51X
!*
CM,_Y1,LINE
CMSEL,,_Y
LESIZE,_Y1, , ,2,1,
CMDEL,_Y
CMDEL,_Y1
! connector piece
FLST,5,1,4,ORDE,1
FITEM,5,163
CM,_Y,LINE
```

```
LSEL, , , ,P51X
!*
CM,_Y1,LINE
CMSEL,,_Y
LESIZE,_Y1, , ,1,1,
CMDEL,_Y
CMDEL,_Y1
!*
FLST,5,24,4,ORDE,2
FITEM,5,166
FITEM,5,-189
CM,_Y,LINE
LSEL, , , ,P51X
!*
CM,_Y1,LINE
CMSEL,,_Y
LESIZE,_Y1, , ,1,1,
CMDEL,_Y
CMDEL,_Y1
! mesh new lines as appropriate
! solid alumiunum
TYPE,  1
MAT,     1
REAL,     8
ESYS,     0
SECNUM, ,
! mesh lines
FLST,2,5,4,ORDE,3
FITEM,2,101
FITEM,2,162
FITEM,2,-165
LMESH,P51X
! mesh the sensor and piezo
TYPE,  3
MAT,     2
REAL,     1
ESYS,    11
SECNUM, ,
! mesh as one hexahedron
MSHAPE,0,3D
MSHKEY,1
!*
FLST,5,2,6,ORDE,2
FITEM,5,1
FITEM,5,-2
CM,_Y,VOLU
VSEL, , , ,P51X
CM,_Y1,VOLU
CHKMSH,'VOLU'
CMSEL,S,_Y
!*
VMESH,_Y1
!*
CMDEL,_Y
CMDEL,_Y1
CMDEL,_Y2
! define the rigid regions at ends
CERIG,P51X, ,ALL, , , ,
CERIG,P51X, ,ALL, , , ,
CERIG,P51X, ,ALL, , , ,
CERIG,P51X, ,ALL, , , ,
! change piezo to PZT material
EMODIF,P51X,MAT,2,
! define material for quartz
UIMP,3,EX,EY,EZ,100000000000,
100000000000, 100000000000,
UIMP,3,DENS, , ,23000,
UIMP,3,KXX,KYY,KZZ,1,1,1,
UIMP,3,MURX,MURY,MURZ,1,1,1,
UIMP,3,RSVX,RSVY,RSVZ,1,1,1,
UIMP,3,PERX,PERY,PERZ,4.52,4.52,4.68,
! define material for PZT
UIMP,3,EX,EY,EZ,64.5e9, 64.5e9, 64.5e9,
UIMP,3,DENS, , ,7600,
UIMP,3,KXX,KYY,KZZ,1,1,1,
UIMP,3,MURX,MURY,MURZ,1,1,1,
UIMP,3,RSVX,RSVY,RSVZ,1,1,1,
UIMP,3,PERX,PERY,PERZ,1730,1730,1700,
! activate PIEZ for PZT
TB,PIEZ,2, , , ,
TBMODIF,3,3,7100
! activate PIEZ for quartz
TB,PIEZ,3, , , ,
TBMODIF,3,3,8756
! finished getting the control elements done
EMODIF,P51X,REAL,8,
! fix the elements on the active strut to material 4
FLST,2,5,2,ORDE,2
FITEM,2,671
FITEM,2,-675
EMODIF,P51X,MAT,4,
! unmesh strut with LPACT
LCLEAR,    127
! divide into two lines
LDIV,127, , ,2,0
FLST,5,2,4,ORDE,2
FITEM,5,127
FITEM,5,190
CM,_Y,LINE
LSEL, , , ,P51X
!* define one element per line
CM,_Y1,LINE
CMSEL,,_Y
LESIZE,_Y1, , ,2,1,
CMDEL,_Y
CMDEL,_Y1
!* mesh
TYPE,  1
```

```
MAT,     1
REAL,     1
ESYS,     0
SECNUM, ,
!*
FLST,2,2,4,ORDE,2
FITEM,2,127
FITEM,2,190
LMESH,P51X
! define mass of LPACT
R,9,2.224,
! mesh keypoint`
TYPE,  2
MAT,     1
REAL,     9
ESYS,     0
SECNUM, ,
!*
KMESH,    75
! couple voltage DOF's at SOLID5's
FLST,4,4,1,ORDE,2
FITEM,4,444
FITEM,4,-447
CP,1,VOLT,P51X
FLST,4,4,1,ORDE,2
FITEM,4,452
FITEM,4,-455
CP,2,VOLT,P51X
FINISH
/SOLU
! define damping constants
ALPHAD,9.1,
BETAD,1.1e-3,
DMPRAT,0,
! modified truss complete
! *
```

APPENDIX I. ACT_TRUSS.INP

```
! ************************************************
! * act_truss_3.inp                              *
! * used in getting the fresh truss on line      *
! * Truss Control APDL Program for ANSYS version 5.5 *
! * For simple sinusoidal disturbances           *
! * Applied to the NPS Space Truss Active Controlled Model *
! * Written by LT Carey M. Pantling              *
! * Last Modified 14 Oct 1999                    *
! ************************************************

! First Load Truss model with mesh and BC's

! ********************
! * Define Variables     *
! ********************

eplot                   ! gives something to look at while waiting

! *set command will define and clear variables
*set,FREQ,11.75         ! disturbance frequency
*set,CS,0.0             ! control signal to piezo, initial zero
*set,PV,68              ! Piezo voltage node# for application
*set,RS,                ! read signal from piezo, defined only
*set,SV,60              ! Sensor voltage node# for detection
*set,G1,300             ! Control gain constants, see below
*set,G2,100
*set,G3,0.0
*set,OLDRS,0.0          ! the old RS, IC 0.0
*set,OLDINT,0.0         ! old integral, IC 0.0
*set,OLDFINT,0.0        ! old filtered integral
*set,OLDDBL,0.0         ! prev DI, IC 0.0
*set,OLDFDBL,0.0        ! old filtered DI
*set,STRTTIME,0.0       ! Start time = 0.0 seconds

! set to more reasonable later
*set,FINITIME,4.0       ! Finish Time in seconds

TIMESTEP=1/FREQ/20      ! TIMESTEP
*set,PI,acos(-1)
*set,DISPNODE,25        ! node 26 real truss for reading the output
*set,APLNODE,53         ! the node where the LPACT is located
finish
```

```
! ********************************************************
! * Set up load disturbance, at least 20 steps per period *
! ********************************************************

*set,NUMSTEP,nint(FINITIME/TIMESTEP)
*set,FORCFUN,
*set,FFP,
*dim,FORCFUN,array,NUMSTEP          ! array for forcing function
*dim,FFP,table,NUMSTEP              ! table for plotting
! arrays for input (indexed), tables for output(non indexed)

! create a magnitude based upon the frequency, from the LPACT chart.
*set,MAG,0.05437*FREQ**2-1.8874*FREQ+18.2439  ! (N) force from LPACT

! Make a table for the force disturbance
*do,ICOUNT,1,NUMSTEP
 FORCFUN(ICOUNT)=sin(2*PI*FREQ*((ICOUNT-1)*TIMESTEP))*MAG
 FFP(ICOUNT)=sin(2*PI*FREQ*((ICOUNT-1)*TIMESTEP))*MAG
*enddo

*set,NODE26Y
*dim,NODE26Y,table,NUMSTEP   ! creates an array to get displacement
*set,CSCheck                 ! tracks the control output signal
*dim,CSCheck,table,NUMSTEP   ! gets the control signal
*set,RSCheck                 ! tracks the read signal
*dim,RSCheck,table,NUMSTEP   ! copies the sensed signal

! these may be deleted, and not be required to be copied, included for error checking
! *set,INTcheck
! *dim,INTcheck,table,NUMSTEP
! *set,DBLcheck
! *dim,DBLcheck,table,NUMSTEP
! *set,FINTchk
! *dim,FINTchk,table,NUMSTEP
! *set,FDBLchk
! *dim,FDBLchk,table,NUMSTEP

! ***************************************
! * Do loop for loading at each time step *
! ***************************************

*do,ICOUNT,1,NUMSTEP
! Set current time (actually, end of current step)
 *set,CURRTIME,STRTTIME+ICOUNT*TIMESTEP
```

```
 /solu                                  ! enter the solution processor
 *if,ICOUNT,eq,1,then
  antype,trans,new                      ! Start new transient analysis
 *else
  antype,trans,rest                     ! Restart or continue transient analysis
 *endif
 time,CURRTIME                          ! set current time for ANSYS solution
 deltim,TIMESTEP                        ! set delta-t
 f,APLNODE,fy,FORCFUN(ICOUNT)*0.707     ! apply force to APLNODE node
(LPACT)
 f,APLNODE,fz,FORCFUN(ICOUNT)*0.707     ! apply force to APLNODE node
(LPACT)
 d,PV,volt,CS                           ! apply CS voltage to PV node
 allsel,all                             ! select everything to prepare for solve
 solve
 finish
 *if,ICOUNT,gt,200,then                 ! turns on control at specified step
  G3=20.0
 *endif

 /post1                                 ! enter post processor
 *get,RS,node,SV,volt                   ! get sensor voltage
 RSCheck(ICOUNT)=RS                     ! saves current RS in table
 *get,NODE26Y(ICOUNT),node,DISPNODE,u,y ! get the tip displacement
 finish

 ! ****************************************
 ! * Control Law implementation, for next time step   *
 ! ****************************************

 *set,INT,OLDINT+(RS+OLDRS)/2*TIMESTEP          ! approx the integral
 *set,FINT,0.95*OLDFINT+INT-OLDINT              ! digital high pass filter
 *set,DBLINT,OLDDBL+(FINT+OLDFINT)/2*TIMESTEP   ! second integral
 *set,FDBL,0.95*OLDFDBL+DBLINT-OLDDBL          ! digital high pass filter!
! INTcheck(ICOUNT)=INT                         ! copies the terms for examination
! FINTchk(ICOUNT)=FINT
! DBLcheck(ICOUNT)=DBLINT
! FDBLchk(ICOUNT)=FDBL
 OLDRS=RS                                      ! cycle the terms to old values
 OLDINT=INT
 OLDFINT=FINT
 OLDDBL=DBLINT
 OLDFDBL=FDBL
 CS=(G1*FINT+G2*FDBL)*G3                ! CS=RS*(G1/s+g2/s^2)*G3
```

```
  *if,CS,gt,60,then                    ! voltage limiter
   CS=60
  *endif
  *if,CS,lt,-60,then
   CS=-60
  *endif
  CSCheck(ICOUNT)=CS                   ! save for output

*enddo                                 ! ends the loop, ready for next time step

! ************************
! * Plot the results to screen *
! ************************
/erase
*vplot,,NODE26Y(1)

! ***********************************************
! * Write RSCheck results to text file "ATR_e#_r#.out" *
! *   and NODE26Y results to ATN.                *
! *   and CSCheck results to ATS.                *
! ***********************************************
*cfopen,AR_e1_r1,out,
*vwrite,CSCheck(1),
(E10.3)
*cfclos
*cfopen,AN_e1_r1,out,
vwrite,NODE26Y(1),
(E10.3)
*cfclos
*cfopen,AS_e1_r1,out,
vwrite,NODE26Y(1),
(E10.3)
*cfclos
```

APPENDIX J. NPS_PREP.M

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% nps_prep.m
% simulink prep program by LT Carey M. Pantling\
% places one active member in the structure
% generates the inverse matrices and so on
% variables needed:
%     Apply at node 3    aan3
%     InvMass            Im
%     InvMass*C          Imc
%     InvMass*K          Imk
%     Application MatrixAM
%     Output Select      OS
%
% run Final_root_nps, then have the following variables in workspace
%
%     kk    global stiffness matrix
%     mm    global mass matrix
%     nd    nodal connections for elements (nel by 2)
%     coord coordinate values for the number of nodes (nel by 3)
%     nnode number of nodes in system
%     ndof  number of degrees of freedom per node
%     sdof  system degrees of freedom
%     bcval,bcdof boundary conditions
%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% get the application matrix for the active element

cel=101;    % element that has active strut
% compute elemetal dimensional data
dx=coord(nd(cel,2),1)-coord(nd(cel,1),1);
dy=coord(nd(cel,2),2)-coord(nd(cel,1),2);
dz=coord(nd(cel,2),3)-coord(nd(cel,1),3);
leng=(dx^2+dy^2+dz^2)^0.5;

% compute element rotation matrix
%   local aligned with x along axis, y,z orthogonal
%   since tube segments, y,z do not matter
% DCM defined as {x}g=g[c]l*{x}l; g[c]l is the transformation matrix
ul=[(dx/leng),(dy/leng),(dz/leng)];
cr=cross(ul,[0,0,1]);
mag=(cr(1)^2+cr(2)^2+cr(3)^2)^0.5;
if mag~=0 % if {ul}x{i}~=0 then
   OA=[0,1,0;-1,0,0;0,0,1]*ul'; % rotate z-90degrees
   else
      OA=[0,0,-1;0,1,0;1,0,0]*ul'; % rotate y-90 degrees
end
vl=cross(ul,OA); vl=vl/(vl(1)^2+vl(2)^2+vl(3)^2)^0.5;
wl=cross(ul,vl); wl=wl/(wl(1)^2+wl(2)^2+wl(3)^2)^0.5;
DCM=[ ul(1) , vl(1) , wl(1);...
      ul(2) , vl(2) , wl(2);...
```

```
      ul(3) , vl(3) , wl(3) ];
r=zeros(12);
r(1:3,1:3)=DCM;
r(4:6,4:6)=DCM;
r(7:9,7:9)=DCM;
r(10:12,10:12)=DCM;
% gcl=r lcg=r'

am=zeros(12,1);
am(1)=-1; am(7)=1;% axial transfer matrix
am=r*am; % app matrix=gcl*atm

isom=zeros(1,12);
isom(1)=-1; isom(7)=1;
isom=isom*r';

AM=zeros(sdof,1); % Initialize the Application matrix
ISOM=zeros(1,sdof); % Initialize the isolator matrix
start=(nd(cel,2)-1)*6+1;
AM(start:start+5)=am(1:6);
ISOM(start:start+5)=isom(1:6);
start=(nd(cel,1)-1)*6+1;
AM(start:start+5)=am(7:12);
ISOM(start:start+5)=isom(7:12);

% [m]{a}+[c]{v}+[k]{x}={f}
% {a}+inv[m][c]{v}+inv[m][k]{x}=inv[m]{f}
% setting BC's sets imk values to 1.0 the imf values to BCval
% (s^2[m]+s[c]+[k]){x}={f}

% Inverse Mass
% requires squishing the mass matrix to the displacement terms
inverting, expanding
msq=mm; % preserves the original mass matrix
for index=nnode:-1:1
   % remove the back terms first so as to not let the matrix be confused
   start=(index-1)*6+4;
   fini=start+2;
   msq(start:fini,:)=[];
   msq(:,start:fini)=[];
end
invmsq=inv(msq);
% repopulate the Im matrix
Im=zeros(sdof);
for index=1:nnode
   for j=1:nnode
      start=(index-1)*6+1;    startsq=(index-1)*3+1;
      fini=start+2;           finisq=startsq+2;
      Im(start:fini,start:fini)=invmsq(startsq:finisq,startsq:finisq);
   end
end

% Inverse Mass * C (no natural damping)
Imc=zeros(sdof);
```

```
% Inverse Mass * K
Imk=Im*kk;

% apply the boundary conditions, doesn't matter since all zero
% [Imk,Im,ff]=feaplyc2(Imk,Im,ff,bcdof,bcval);

% remove the constrained points from the mass and stiff matrices at
% nodes 1,2,27,28 we're not tracking boundary forces, so we don't care
% what the forces are and can remove from model

sdof=sdof-24;
% reduce mm
Im(157:168,:)=[]; % nodes 27,28
Im(:,157:168)=[];
Im(1:12,:)=[];    % nodes 1,2
Im(:,1:12)=[];
% reduce Imc
Imc(157:168,:)=[]; % nodes 27,28
Imc(:,157:168)=[];
Imc(1:12,:)=[];    % nodes 1,2
Imc(:,1:12)=[];
% reduce kk
Imk(157:168,:)=[]; % nodes 27,28
Imk(:,157:168)=[];
Imk(1:12,:)=[];    % nodes 1,2
Imk(:,1:12)=[];
% reduce AM
AM(157:168)=[]; % nodes 27,28
AM(1:12)=[];    % nodes 1,2
% reduce ISOM
ISOM(157:168)=[]; % nodes 27,28
ISOM(1:12)=[];    % nodes 1,2

% Apply at node 3, (now node 1)
aan3=zeros(sdof,1);
aan3(1:3)=1;

% output select gets y translation of node 3, (now node 1)
OS=zeros(1,sdof);
OS(2)=1;
```

APPENDIX K. HP 35665A DYNAMIC SIGNAL ANALYZER SETUP

The HP 35665A DSA has several different keys that are available to the user to establish a configuration. These keys are of two types hard keys, which do not change value and soft keys, which do change value as various hard keys are pressed. Hard keys will be represented by **[Hard Key]**; soft keys will be represented by [Soft Key]. All the specifics on the commands used are contained in Reference 29. Hewlett-Packard, *HP35665A Dynamic Signal Analyzer Operator's Reference.*

Step 1: Turn on the spectrum analyzer.

The 1/0 (power) switch is on the front panel of the machine.

Step 2: Define the analyzer instrument mode.

[Inst Mode] [2 CHANNEL].

Step 3: Define the analyzer's frequency bandwidth.

[Freq] [SPAN] enter 200 Hz,

[RESOLUTION]enter 800 Lines, and

[RECORD LENGTH] enter 4 Seconds.

Step 4: Select the Display Format.

[Disp Format] [UPPER/LOWER].

Step 5: Display frequency magnitude response on trace A.

[Meas Data] [FREQUENCY RESPONSE].

[Trace Coord] [MAGNITUDE].

Step 6: To Display frequency phase response on trace B.

[Active Trace] will toggle the active trace to trace B

[FREQUENCY RESPONSE].

Step 7: Define Trace B coordinates to phase.

[Trace Coord] [PHASE].

Step 8: Establish Random noise Source.

[Source] [LEVEL] enter in 1.00 [Vpk].

[RANDOM NOISE]

[SOURCE ON/OFF] toggles on the active source

Step 9: Create Frequency Resopnse Window.

[Window] [HANNING].

Step 10: Establish Averaging for Random Signal.

[AVG] [VECTOR] preserves the phase information .

[NUMBER] select 20 average cycles

Step 11: Save data on 3-1/2 inch floppy disk.

[Disk/Util] [SAVE DATA]

[SAVE TRACE]

[INTO FILE] type in filename at attached keyboard.

APPENDIX L. DSA_PLOT.M

```
% ******************************************************
% *                                                    *
% * HP 35665A Data processing utility                  *
% * Designed for the NPS Space Truss Experiment        *
% * C:\Space_truss\truss99\Experiments\hp_dsa          *
% * By LT Carey M. Pantling, USN                       *
% * Last Modified 8/6/99                               *
% *                                                    *
% ******************************************************

cd c:\space_truss\truss99\experiments\hp_dsa

load d99_051a.mat       % load the data file from the HP-35665A
      % creates the following variables
      % o2i1      the vector of nlines+1 of complex results
      % o2i1x0
      % o2i1xi    frequency resolution
fstart=0;
o2i1xi      % print the freq resolution to WS
fend=50;

freq=[fstart:o2i1xi:fend];
figure(1)

subplot(2,1,1);
      semilogy(freq,abs(o2i1));                % plots the magnitude
   axis([fstart,fend,1e-4,1]);
   xlabel('Frequency (Hz)');
   ylabel('Magnitude');
   title('Magnitude and Phase plots (0-50 Hz)');
   grid;

subplot(2,1,2);
   ph=atan2(imag(o2i1),real(o2i1)); % creates phase +/- pi
      plot(freq,ph*180/pi);                    % plots phase in degrees
      axis([fstart,fend,-180,180]);
      xlabel('Frequency (Hz)');
   ylabel('Phase (deg)');
   grid;
```

APPENDIX M. ACQ_DATA.M

```
% ************************************************
% *                                              *
% * Data Acquisition subroutine for dSPACE       *
% * Designed for the NPS Space Truss Experiment  *
% * By LT Carey M. Pantling, USN                 *
% * Located C:\Space_truss\Experiments\Experiment *
% * Last Modified 8/23/99                        *
% *                                              *
% ************************************************

% verify that the correct directory is being used
cd c:\space_truss\truss99\experiments\experiment_2

% select DS1103 board for use with MLIB and MTRACE
mlib('SelectBoard','DS1103');
mtrc1103('SelectBoard','DS1103');

% check if the application ST_controller_3.ppc is running on dSPACE DSP
DemoApplName = lower(['c:\space_truss\truss99\st_controller_3.ppc']);
if mlib('IsApplRunning'),
   ApplInfo = mlib('GetApplInfo');
   if strcmp(DemoApplName,lower(ApplInfo.name)) ~= 1
      err_msg = sprintf('***This m-file needs the PPC `,...
         `application\n*** ''%s'' running!', DemoApplName);
      error(err_msg);
   end;
else
   err_msg = sprintf('*** This m-file needs the PPC `,...
         `application\n*** ''%s',' running!',DemoApplName);
   error(err_msg);
end;

% select variables to be traced and obtain their descriptors
var_names ={'Model Root/B:input->ADC'; ...
                  'Model Root/B:Sum'; ...
                  'Model Root/S:Accelerometer 1(1)'; ...
                  'Model Root/S:Accelerometer 1(2)'; ...
                  'Model Root/S:Accelerometer 1(3)'; ...
                  'Model Root/S:Accelerometer 2(1)'; ...
                  'Model Root/S:Accelerometer 2(2)'; ...
                  'Model Root/S:Accelerometer 2(3)'; ...
                  'Model Root/S:Accelerometer 3(1)'; ...
                  'Model Root/S:Accelerometer 3(2)'; ...
                  'Model Root/S:Accelerometer 3(3)'; ...
                  'Model Root/S:Accelerometer 4(1)'; ...
                  'Model Root/S:Accelerometer 4(2)'; ...
                  'Model Root/S:Accelerometer 4(3)'};

var = mlib('GetTrcVar',var_names);

% set data acquisition to be performed by service number 1 (default)
```

```
ST=20.0;            % sample time (sec)
FS=500;             % sample Frequency (Hz), set by simulink parameters
NS=ST*FS;    % number of samples

mtrc1103('Set','TraceVars', var, 'NumSamples',NS);

mtrc1103('StartCapture');                % start capture on DS1103
while mtrc1103('CaptureState')~=0,end  % wait until capture is done

% Fetch after capture is complete
out_data = mtrc1103('FetchData');

% Plot code extensively modified from 'Graph.m'
% originally written by LT John Vlattas and LT Scott Johnson 5/10/98

timevec=[1/FS:ST/NS:ST];
for n = 1:14
      eval(['Y' num2str(n) ' = out_data(n,:);']);
      % Separates Data to vectors
   if n < 3
            figure(1)
            subplot(2,1,n)
                  if n == 1
                        plot(timevec,eval(['Y' num2str(n)]))
                        title('PCB Force Sensor - Time Data')
                        xlabel('Time (msec)')
                        ylabel('Amplitude (mV)')
                  elseif n == 2
                        plot(timevec,eval(['Y' num2str(n)]))
                        title('Output of Controller - Time Data')
                        xlabel('Time (msec)')
                        ylabel('Amplitude (mV)')
                  end
            orient tall
   elseif n < 6
            figure(2)
            subplot(3,1,(n-2))
                  if n == 3
                        plot(timevec,eval(['Y' num2str(n)]),'m')
                        title('Node 41 - X-Axis - Time Data')
                        xlabel('Time (msec)')
                        ylabel('Amplitude (mV)')
                  elseif n == 4
                        plot(timevec,eval(['Y' num2str(n)]),'m')
                        title('Node 41 - Y-Axis - Time Data')
                        xlabel('Time (msec)')
                        ylabel('Amplitude (mV)')
                  elseif n == 5
                        plot(timevec,eval(['Y' num2str(n)]),'m')
                        title('Node 41 - Z-Axis - Time Data')
                        xlabel('Time (msec)')
                        ylabel('Amplitude (mV)')
                  end
      orient tall
      elseif n < 9
```

```
        figure(3)
        subplot(3,1,(n-5))
              if n == 6
                    plot(timevec,eval(['Y' num2str(n)]),'m')
                    title('Node 18 - X-Axis - Time Data')
                    xlabel('Time (msec)')
                    ylabel('Amplitude (mV)')
              elseif n == 7
                    plot(timevec,eval(['Y' num2str(n)]),'m')
                    title('Node 18 - Y-Axis - Time Data')
                    xlabel('Time (msec)')
                    ylabel('Amplitude (mV)')
              elseif n == 8
                    plot(timevec,eval(['Y' num2str(n)]),'m')
                    title('Node 18 - Z-Axis - Time Data')
                    xlabel('Time (msec)')
                    ylabel('Amplitude (mV)')
              end
        orient tall
elseif n < 12
        figure(4)
        subplot(3,1,(n-8))
              if n == 9
                    plot(timevec,eval(['Y' num2str(n)]),'m')
                    title('Node 49 - X-Axis - Time Data')
                    xlabel('Time (msec)')
                    ylabel('Amplitude (mV)')
              elseif n == 10
                    plot(timevec,eval(['Y' num2str(n)]),'m')
                    title('Node 49 - Y-Axis - Time Data')
                    xlabel('Time (msec)')
                    ylabel('Amplitude (mV)')
              elseif n == 11
                    plot(timevec,eval(['Y' num2str(n)]),'m')
                    title('Node 49 - Z-Axis - Time Data')
                    xlabel('Time (msec)')
                    ylabel('Amplitude (mV)')
              end
        orient tall
elseif n < 15
        figure(5)
        subplot(3,1,(n-11))
              if n == 12
                    plot(timevec,eval(['Y' num2str(n)]),'m')
                    title('Node 26 - X-Axis - Time Data')
                    xlabel('Time (msec)')
                    ylabel('Amplitude (mV)')
              elseif n == 13
                    plot(timevec,eval(['Y' num2str(n)]),'m')
                    title('Node 26 - Y-Axis - Time Data')
                    xlabel('Time (msec)')
                    ylabel('Amplitude (mV)')
              elseif n == 14
                    plot(timevec,eval(['Y' num2str(n)]),'m')
                    title('Node 26 - Z-Axis - Time Data')
```

```
                        xlabel('Time (msec)')
                        ylabel('Amplitude (mV)')
                    end
                orient tall
    end
end

% Takes Power Spectral Density of Given Vectors For Comparison

Pxx1=fft(100*Y1);
Pxx1=(real(Pxx1).^2+imag(Pxx1).^2).^0.5;
Pxx1((NS/2+1):NS)=[];          % clears the redundant upper portion
F1=FS/NS*[0:NS/2-1];           % vector for plotting the freqs

% Load the uncontrolled case
load Pxx2                      % PSD vector

% Finds Maximum Value of the PSD For Trial In Question
u = max(10*log10(Pxx2(10:NS/2)))
c = max(10*log10(Pxx1(10:NS/2)))
diff = u - c                   % dB reduction due to control

% define trial number
% trial=6;    % trial number located in workspace

figure(6)
% plot(F1(1:400),10*log10(Pxx1(1:400)));
plot(F1(1:400),10*log10(Pxx2(1:400)),'r.',...
     F1(1:400),10*log10(Pxx1(1:400)),'b')
grid;
% title(['Power Spectral Density - Uncontrolled']);
title(['Power Spectral Density-Controlled vs Uncontrolled (Trial ', ...
       num2str(trial),')']);
ylabel('Power Spectrum Magnitude (dB)');
xlabel('Frequency (Hz)');
legend('Uncontrolled','Controlled');
% save trial information in appropriate file
eval(['save trial',num2str(trial),' Y* Pxx1 F1']);
```

APPENDIX N. DATA_PROC.M

```
% ***********************************************
% * Data Processing routine for dSPACE data     *
% * Designed for the NPS Space Truss Experiment *
% * By LT Carey M. Pantling, USN                *
% * Last Modified 10/16/99                      *
% ***********************************************

ST=20.0;          % sample time (sec)
FS=500;           % sample Frequency (Hz), set by simulink parameters
NS=ST*FS;    % number of samples

% Plot code extensively modified from 'Graph.m'
% originally written by LT John Vlattas and LT Scott Johnson 5/10/98

% trial number stored as variable in workspace
% load trial information from appropriate file
eval(['load trial',num2str(trial),' Y* Pxx1 F1']);

timevec=[1/FS:ST/NS:ST];
for n = 1:14
   if n < 3
            figure(1)
            subplot(2,1,n)
                  if n == 1
                        plot(timevec,eval(['Y' num2str(n)]))
                        title('PCB Force Sensor - Time Data')
                        xlabel('Time (sec)')
                        ylabel('Amplitude (mV)')
                  elseif n == 2
                        plot(timevec,eval(['Y' num2str(n)]))
                        title('Output of Controller - Time Data')
                        xlabel('Time (sec)')
                        ylabel('Amplitude (mV)')
                  end
            orient tall
   elseif n < 6
            figure(2)
            subplot(3,1,(n-2))
                  if n == 3
                        plot(timevec,eval(['Y' num2str(n)]),'m')
                        title('Node 41 - X,Y,Z-Axis - Time Data')
                        ylabel('Amplitude (mV)')
                  elseif n == 4
                        plot(timevec,eval(['Y' num2str(n)]),'m')
                        ylabel('Amplitude (mV)')
                  elseif n == 5
                        plot(timevec,eval(['Y' num2str(n)]),'m')
                        xlabel('Time (sec)')
                        ylabel('Amplitude (mV)')
                  end
      orient tall
```

```
        elseif n < 9
              figure(3)
              subplot(3,1,(n-5))
                    if n == 6
                          plot(timevec,eval(['Y' num2str(n)]),'m')
                          title('Node 18 - X,Y,Z-Axis - Time Data')
                          ylabel('Amplitude (mV)')
                    elseif n == 7
                          plot(timevec,eval(['Y' num2str(n)]),'m')
                          ylabel('Amplitude (mV)')
                    elseif n == 8
                          plot(timevec,eval(['Y' num2str(n)]),'m')
                          xlabel('Time (sec)')
                          ylabel('Amplitude (mV)')
                    end
              orient tall
    elseif n < 12
              figure(4)
              subplot(3,1,(n-8))
                    if n == 9
                          plot(timevec,eval(['Y' num2str(n)]),'m')
                          title('Node 49 - X,Y,Z-Axis - Time Data')
                          ylabel('Amplitude (mV)')
                    elseif n == 10
                          plot(timevec,eval(['Y' num2str(n)]),'m')
                          ylabel('Amplitude (mV)')
                    elseif n == 11
                          plot(timevec,eval(['Y' num2str(n)]),'m')
                          xlabel('Time (sec)')
                          ylabel('Amplitude (mV)')
                    end
              orient tall
    elseif n < 15
              figure(5)
              subplot(3,1,(n-11))
                    if n == 12
                          plot(timevec,eval(['Y' num2str(n)]),'m')
                          title('Node 26 - X,Y,Z-Axis - Time Data')
                          ylabel('Amplitude (mV)')
                    elseif n == 13
                          plot(timevec,eval(['Y' num2str(n)]),'m')
                          ylabel('Amplitude (mV)')
                    elseif n == 14
                          plot(timevec,eval(['Y' num2str(n)]),'m')
                          xlabel('Time (sec)')
                          ylabel('Amplitude (mV)')
                    end
              orient tall
    end
end

[Pxx1,F1]=PSD(Y13,2048,FS);    % creates PSD of node 26-Y
% Load the uncontrolled case
load Pxx2 Pxx2
```

```
% Finds Maximum Value of the Power Spectral Density For Trial In
Question
u = max(20*log10(Pxx2(10:100)))
c = max(20*log10(Pxx1(10:100)))
diff = u - c                          % dB reduction due to control

figure(6)
% plot(F1(1:100),20*log10(Pxx1(1:100)));
plot(F1(1:100),20*log10(Pxx2(1:100)),'r-',F1(1:100), ...
      20*log10(Pxx1(1:100)),'b--')
grid;
% title(['Power Spectral Density - Uncontrolled']);
title(['Power Spectral Density - Controlled vs Uncontrolled (Trial ',
...    num2str(trial),')']);
ylabel('Power Spectrum Magnitude (dB)');
xlabel('Frequency (Hz)');
% legend('Uncontrolled',0);
legend('Uncontrolled','Controlled',0);
```

APPENDIX O. EXPERIMENTAL RESULTS

EXPERIMENT 1 12.50 Hz 1.2 Vpp

Trial	IGain	IIGain	SG	Peak (dB)	Reduction	Comments	Plot
0	0	0	0	-46.7128	-	Uncontrolled	**Y**
1	300	0	2.00	-57.8364	11.1236		
2	350	0	2.00	-57.6053	10.8926	Long-term unstable	
3	350	0	1.75	-53.9336	7.2208	Unstable	
4	300	0	1.75	-56.2173	9.5045		
5	300	5000	1.75	-57.7261	11.0173	Rapidly unstable	**Y**
6	300	2500	1.75	-62.8023	16.0895	Unstable in 10 sec	
7	300	1000	1.75	-61.0617	14.3489	Unstable in 15 sec	
8	300	500	1.75	-62.6263	15.9135	Unstable in 20 sec	
9	300	200	1.75	-60.7286	14.0158	Borderline unstable	**Y**
10	300	0	1.75	-62.3806	15.6678	Changed Air Tank	
11	300	100	1.75	-62.3665	15.6537		
12	300	100	2.00	-62.5081	15.7953	Borderline stable	
13	300	-100	1.75	-64.3968	17.6840		
14	300	-200	1.75	Lost	Lost	Data File Lost	
15	300	-400	1.75	-65.2538	**18.5410**	Best Case, this Trial	
16	300	-300	1.75	-65.1145	16.4017		
17	300	100	1.75	-47.4427	0.7299		
18	0	400	1.75	Lost	Lost	Data File Lost	
19	0	1000	1.75	-46.2625	-.4453		
20	0	-1000	1.75	-49.2258	2.5160		
21	0	-2000	1.75	-49.8783	3.1655		
22	0	-4000	1.75	-49.3786	2.6658	Unstable 4 sec	
23	300	-200	1.75	-77.7263	5.5886	1.0 Vpp	
24	0	0	0	-72.1377	-	Uncontrolled 1.0Vpp	
25	300	-200	2.00	-74.7725	2.6348	Borderline stable	
26	300	-200	2.25	-74.9435	2.8058	Borderline stable	
27	300	-200	2.50	-73.5656	1.4279	Borderline stable	
28	300	-200	3.00	-70.9681	-1.1696	Unstable	

Table 19. Experiment 1 Results

EXPERIMENT 1 12.50 Hz 1.2 Vpp TRIAL 0, Uncontrolled

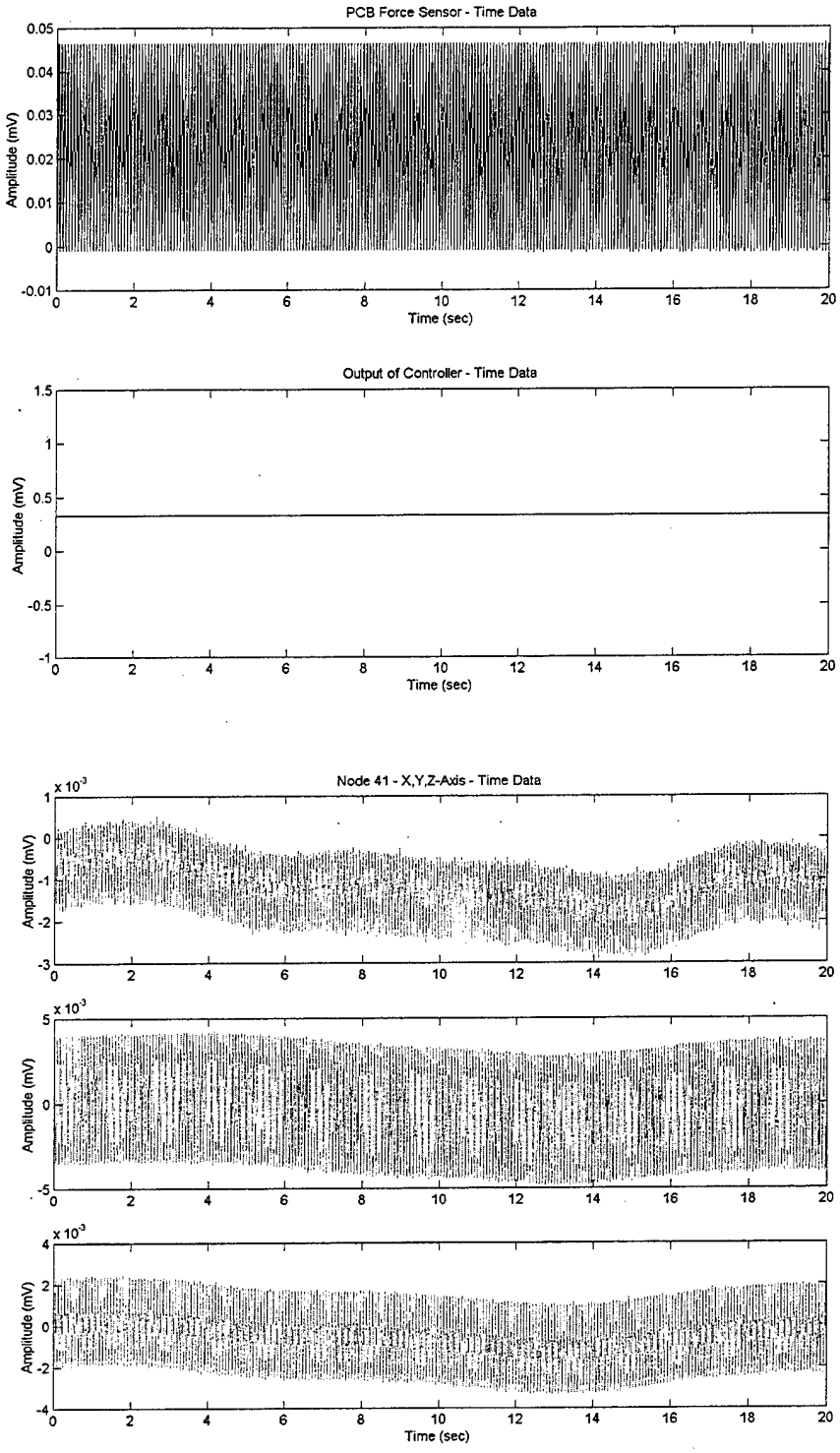


Figure 45. Exp.1 Trial 0 Controller and Node 41 Response

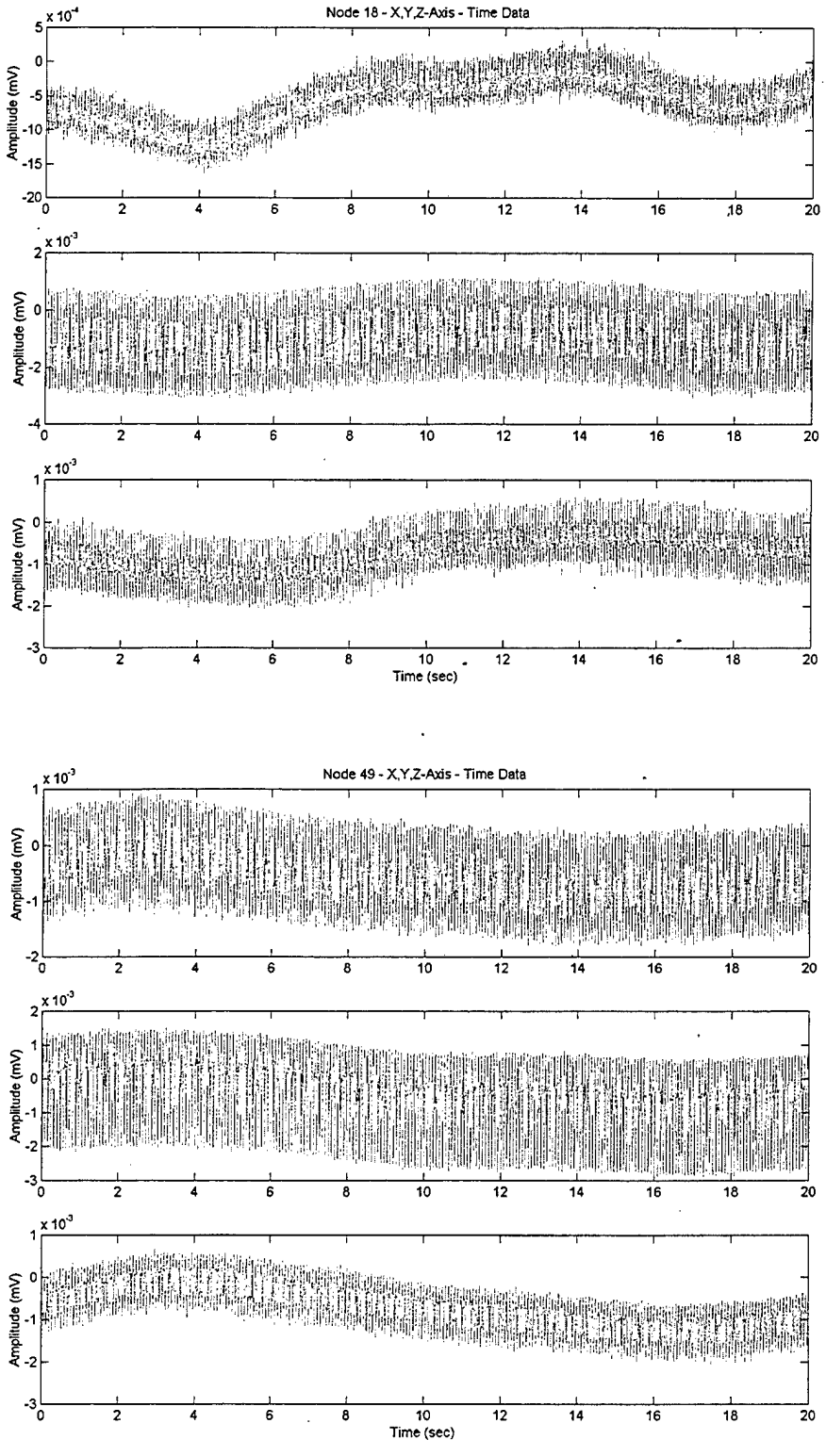
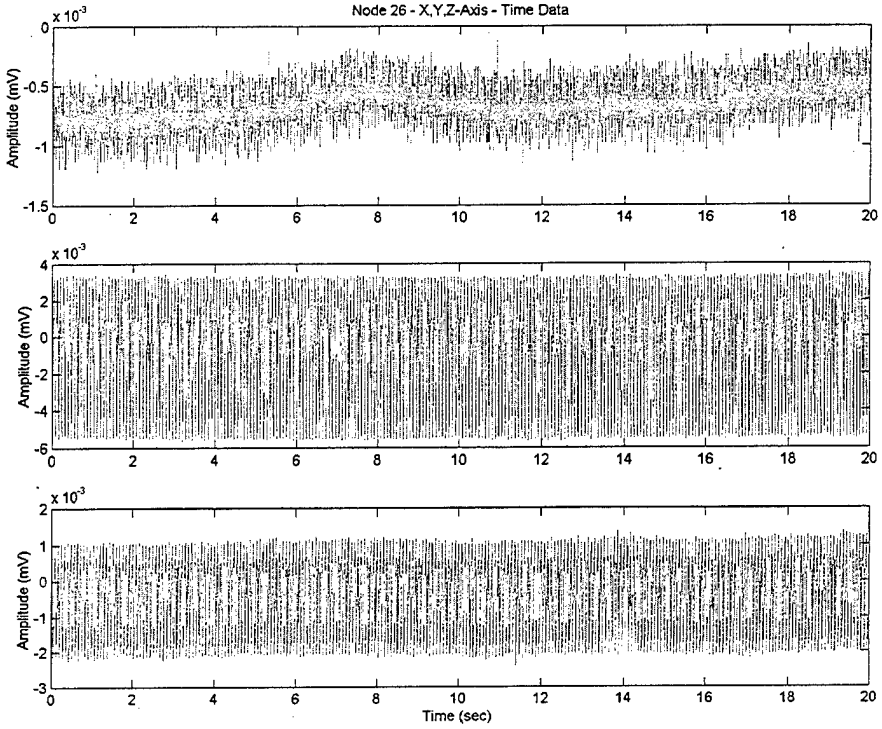


Figure 46. Exp. 1 Trial 0 Node 18 and 49 Response



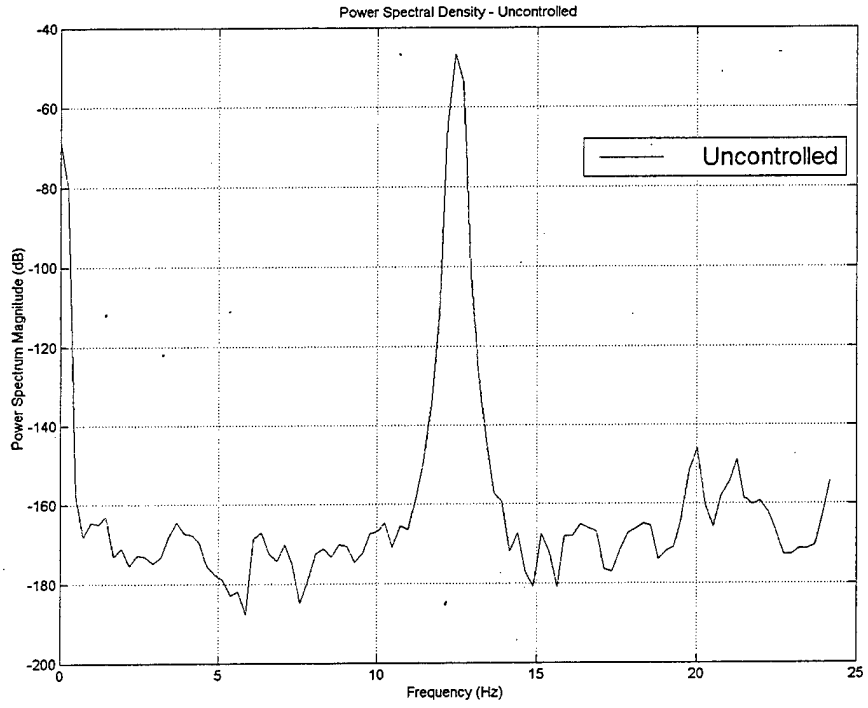


Figure 47. Exp. 1 Trial 0 Node 26 Response and PSD

TRIAL 5: IGain=300, IIGain=5000, SG=1.75

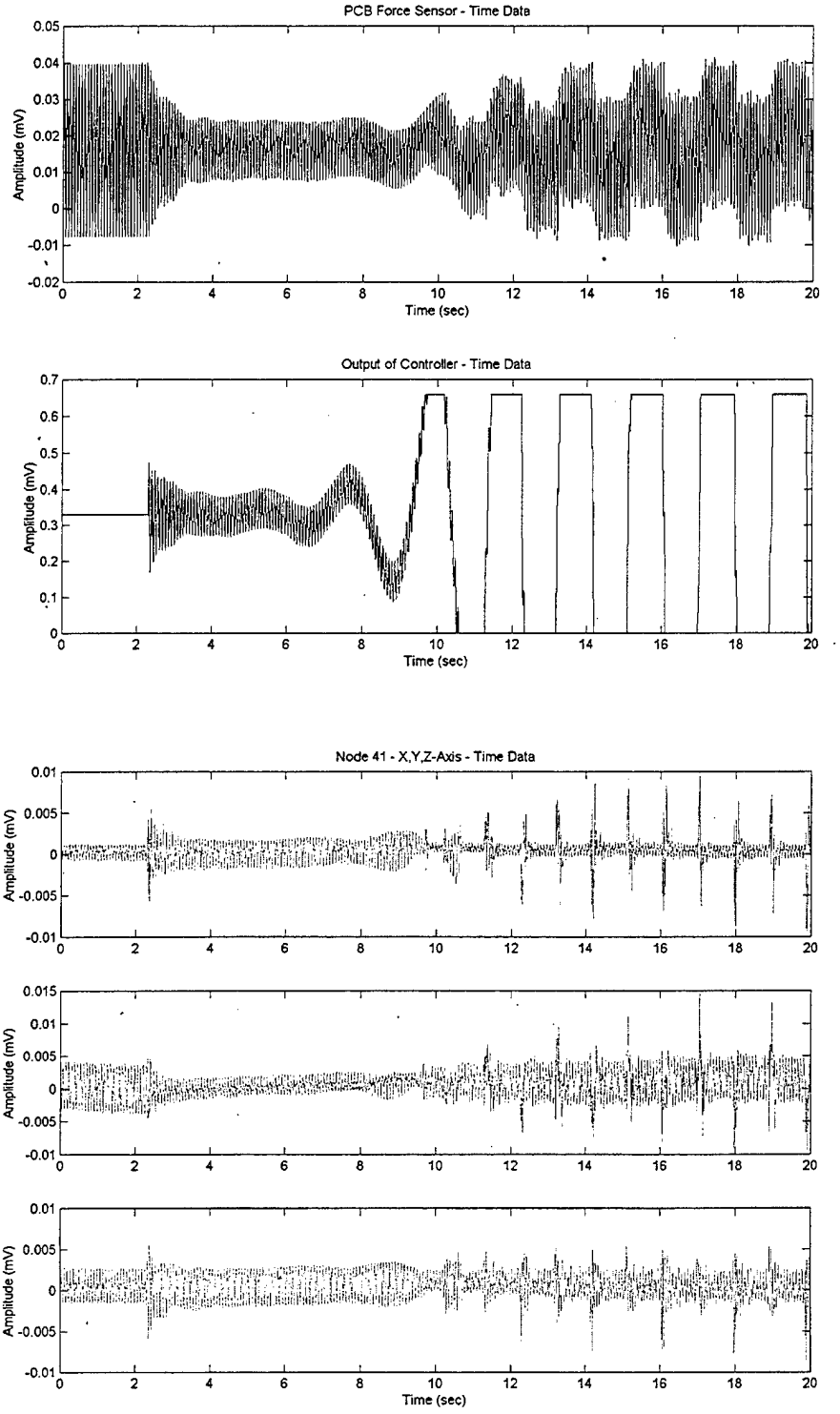


Figure 48. Exp. 1 Trial 5 Controller and Node 41 Response

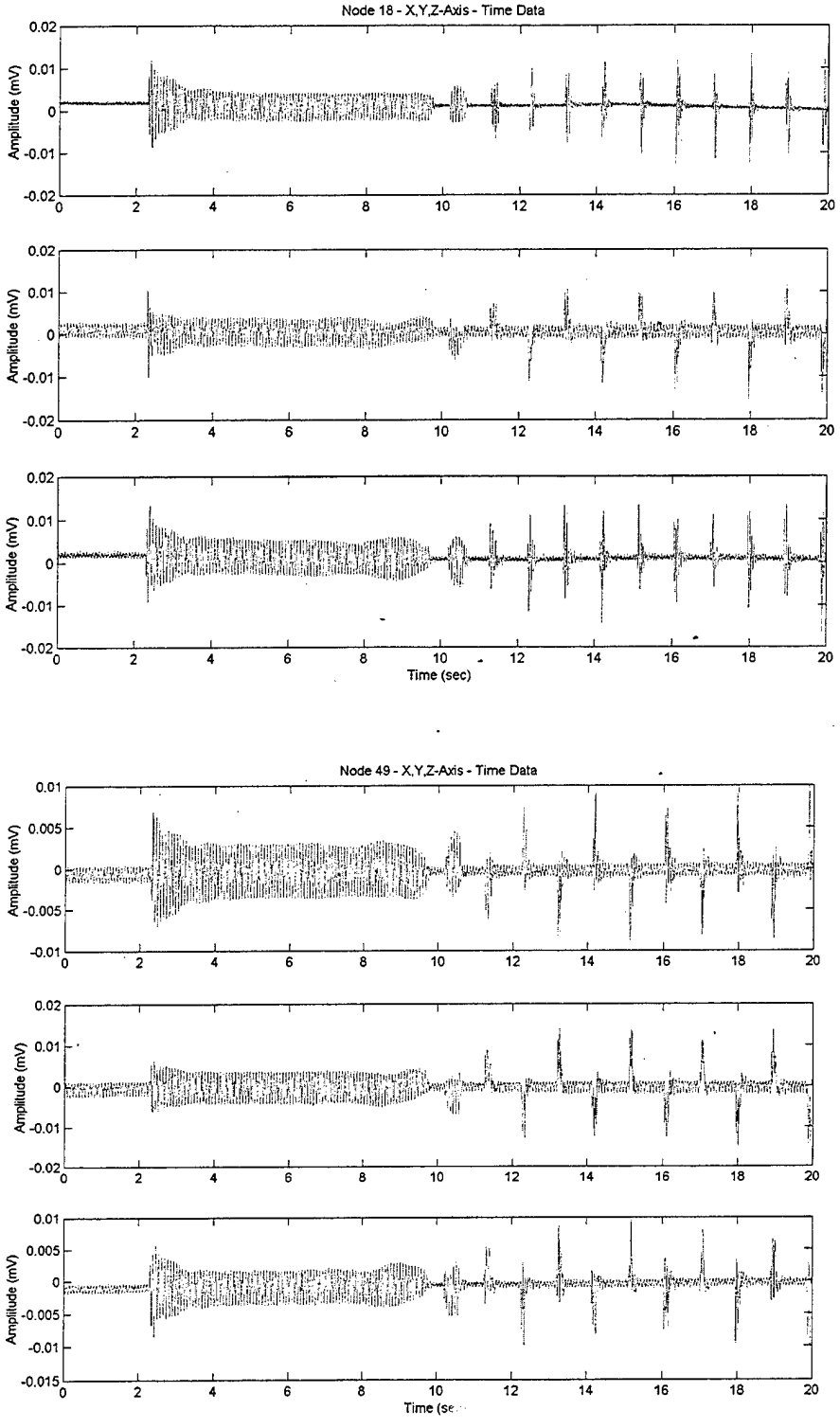


Figure 49. Exp. 1 Trial 5 Node 18 and 49 Response

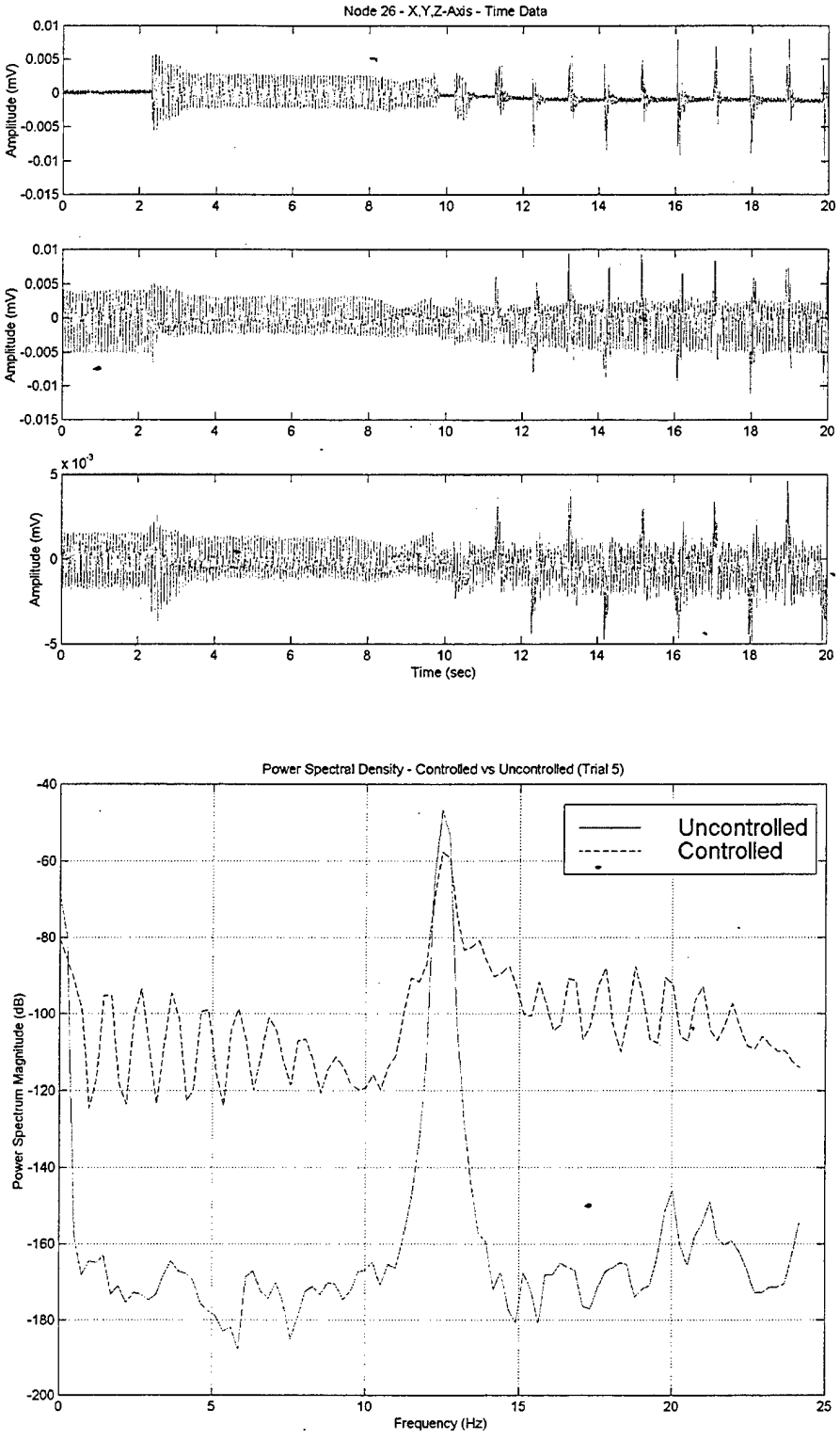


Figure 50. Exp. 1 Trial 5 Node 26 Response and PSD

TRIAL 9: IGain=300, IIGain=200, SG=1.75

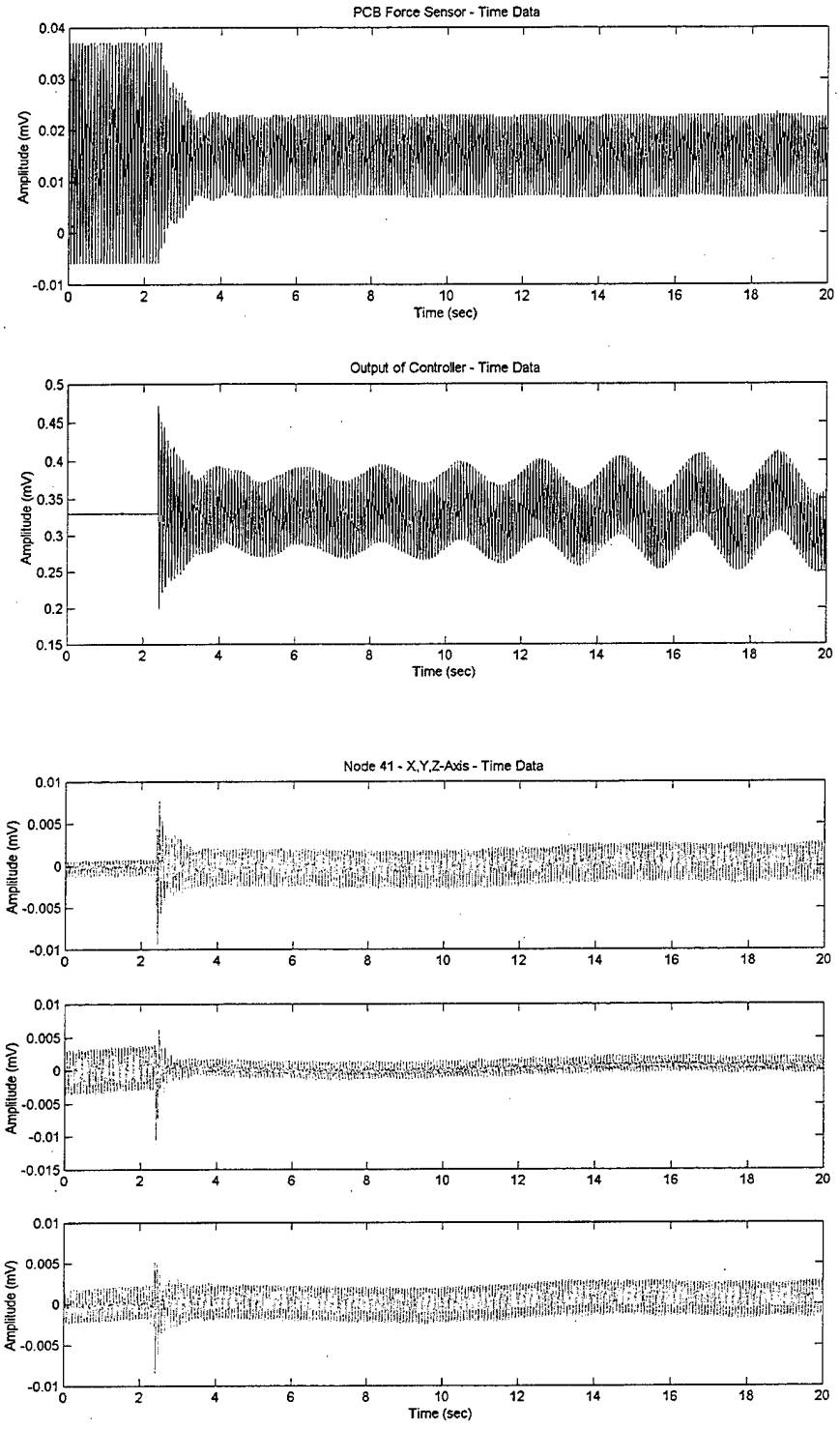


Figure 51. Exp. 1 Trial 9 Controller and Node 41 Response

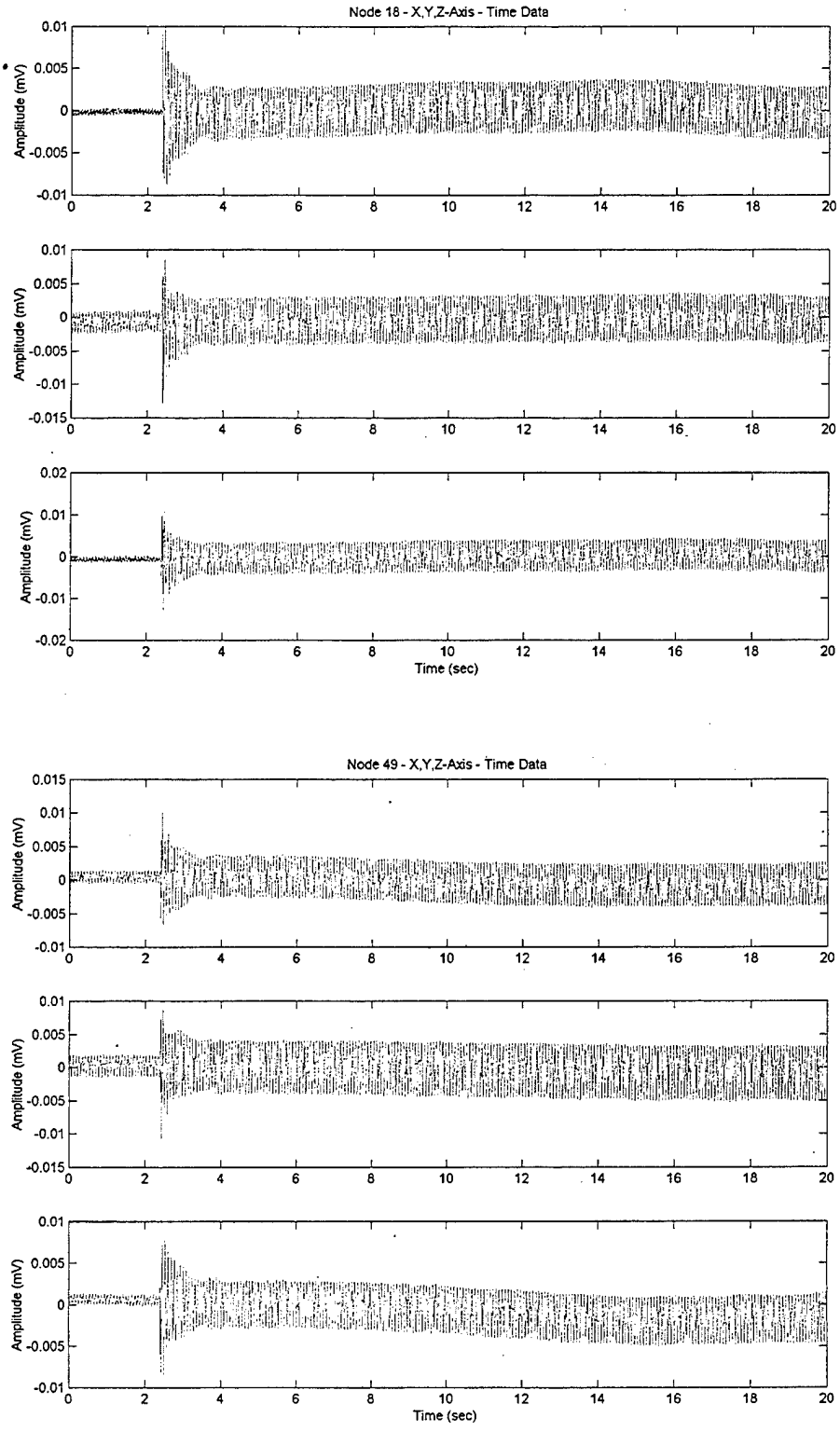


Figure 52. Exp. 1 Trial 9 Node 18 and 49 Response

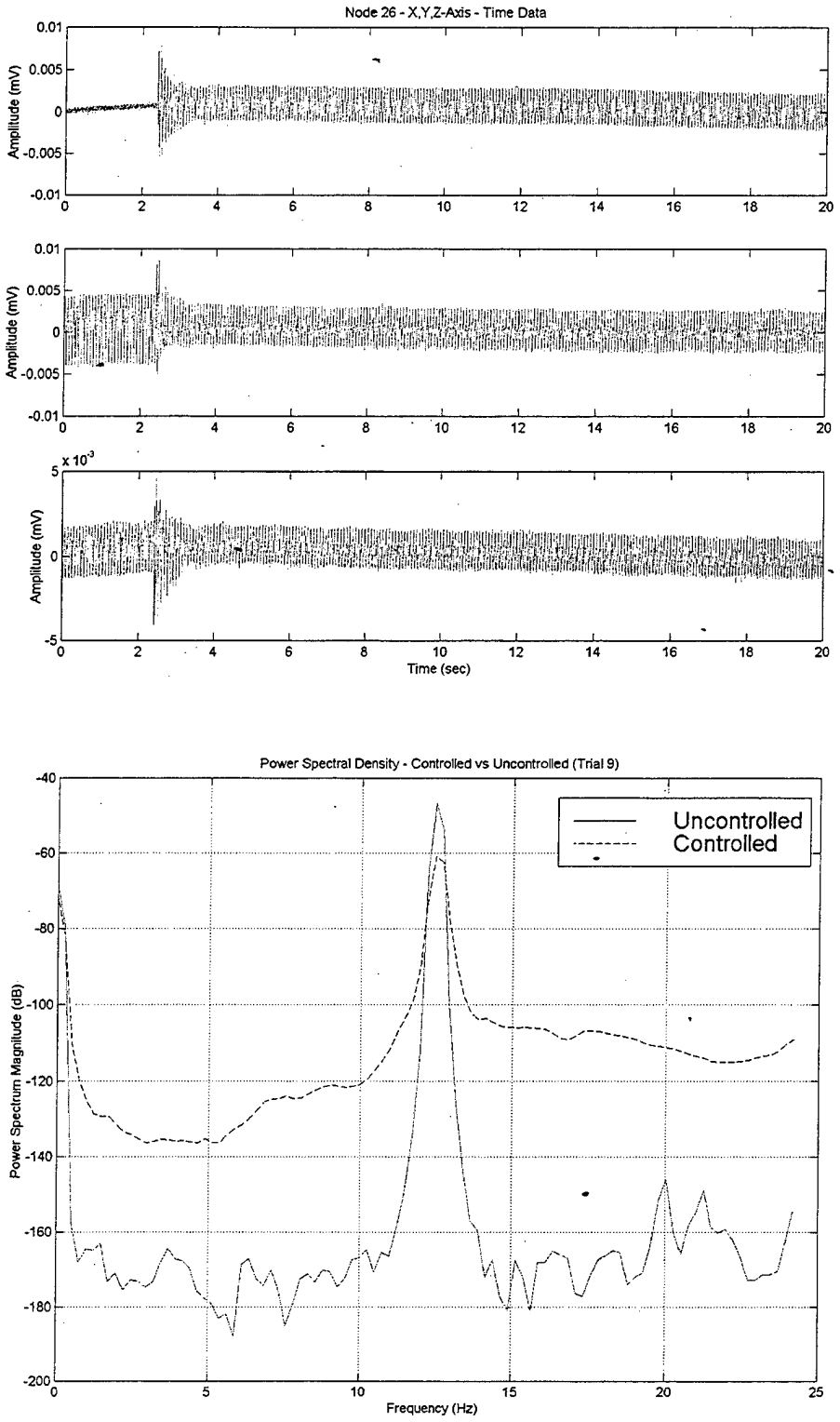


Figure 53. Exp. 1 Trial 9 Node 26 Response and PSD

EXPERIMENT 2: 13.81 Hz 1.2 Vpp

Trial	IGain	IIGain	SG	Peak (dB)	Reduction	Comments	Plot
0	0	0	0	-50.5809	-	Uncontrolled	
1	300	0	2.00	-65.1144	14.5335		
2	350	0	2.00	-67.6224	17.0415	Unstable	
3	350	0	1.75	-68.8184	18.2484	Borderline unstable	
4	300	100	1.75	-68.8293	18.2375	Borderline stable	
5	300	200	1.75	-70.4061	19.8251	Border of stability	
6	300	-100	1.75	-68.5509	17.9700		
7	300	-200	1.75	-73.3524	22.7714	Not Able to Duplicate	
8	300	-300	1.75	-69.6011	**19.0202**	**BEST OVERALL**	**Y**
9	300	-200	2.00	-65.7410	15.1601	Check of Trial 7	
10	300	0	2.00	-66.6074	16.0264		
11	0	200	2.00	-50.0963	-0.4046		
12	0	-200	2.00	-50.2660	-0.3149		
13	0	1000	2.00	-50.4092	-.1717		
14	0	5000	2.00	-50.5151	-.0658	Unstable 9 sec	
15	0	-1000	2.00	-50.1624	-.4185		
16	0	-2000	2.00	-51.1882	-.3927		
17	0	-5000	2.00	-51.5259	0.9450	Unstable 5 sec	
18	0	0	0	-84.3425	-	Uncontrolled 1.0 Vpp	
19	300	0	1.75	-97.7637	13.4212		
20	300	0	2.25	-98.5186	14.1761	Noisy Signal	
21	300	0	3.5	-96.4280	12.0855	Unstable 2 sec	

Table 20. Experiment 2 Results

TRIAL 8: IGain=300, IIGain=-300, SG=2.0
BEST OVERALL CASE

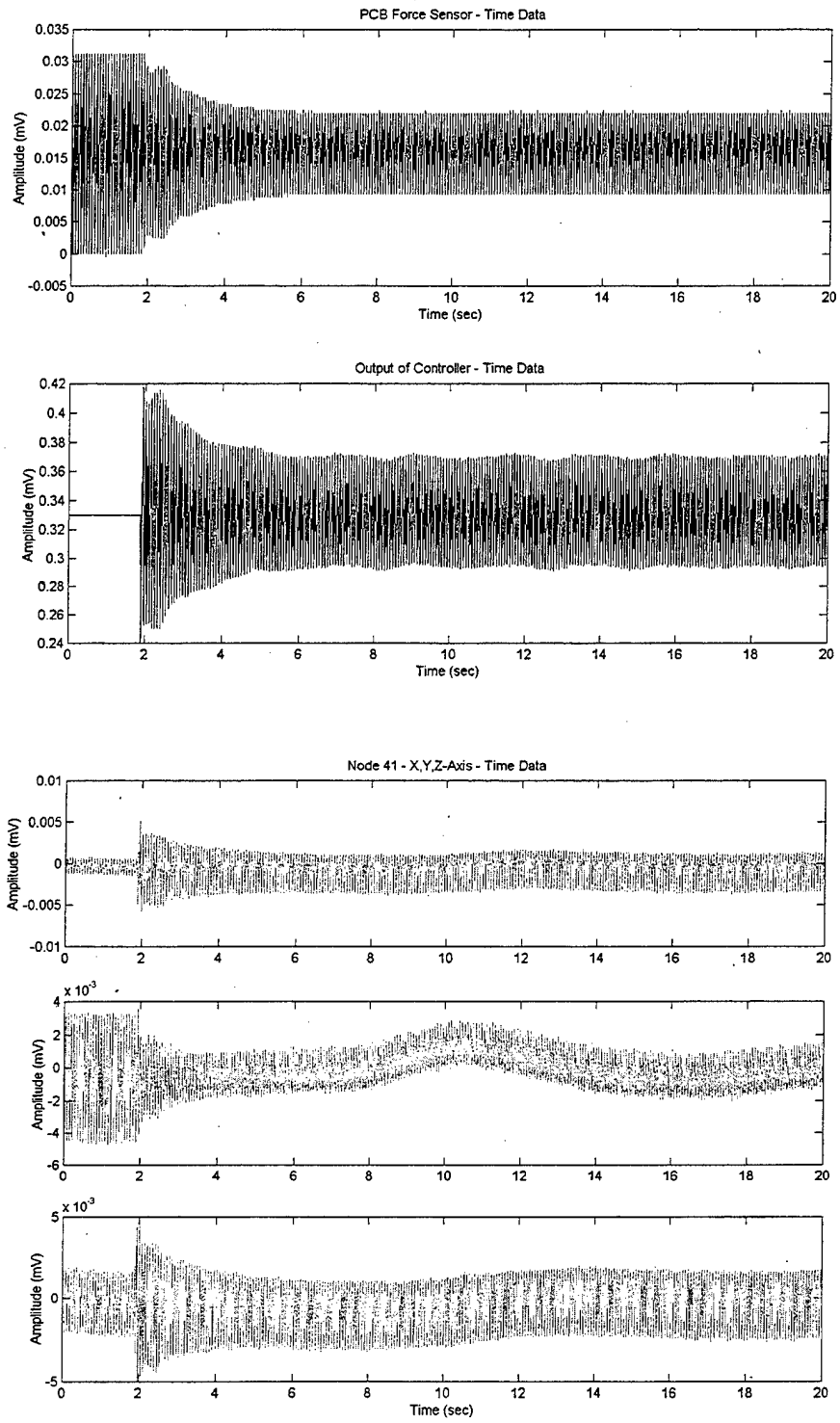


Figure 54. Exp. 2 Trial 8 Controller and Node 41 Response

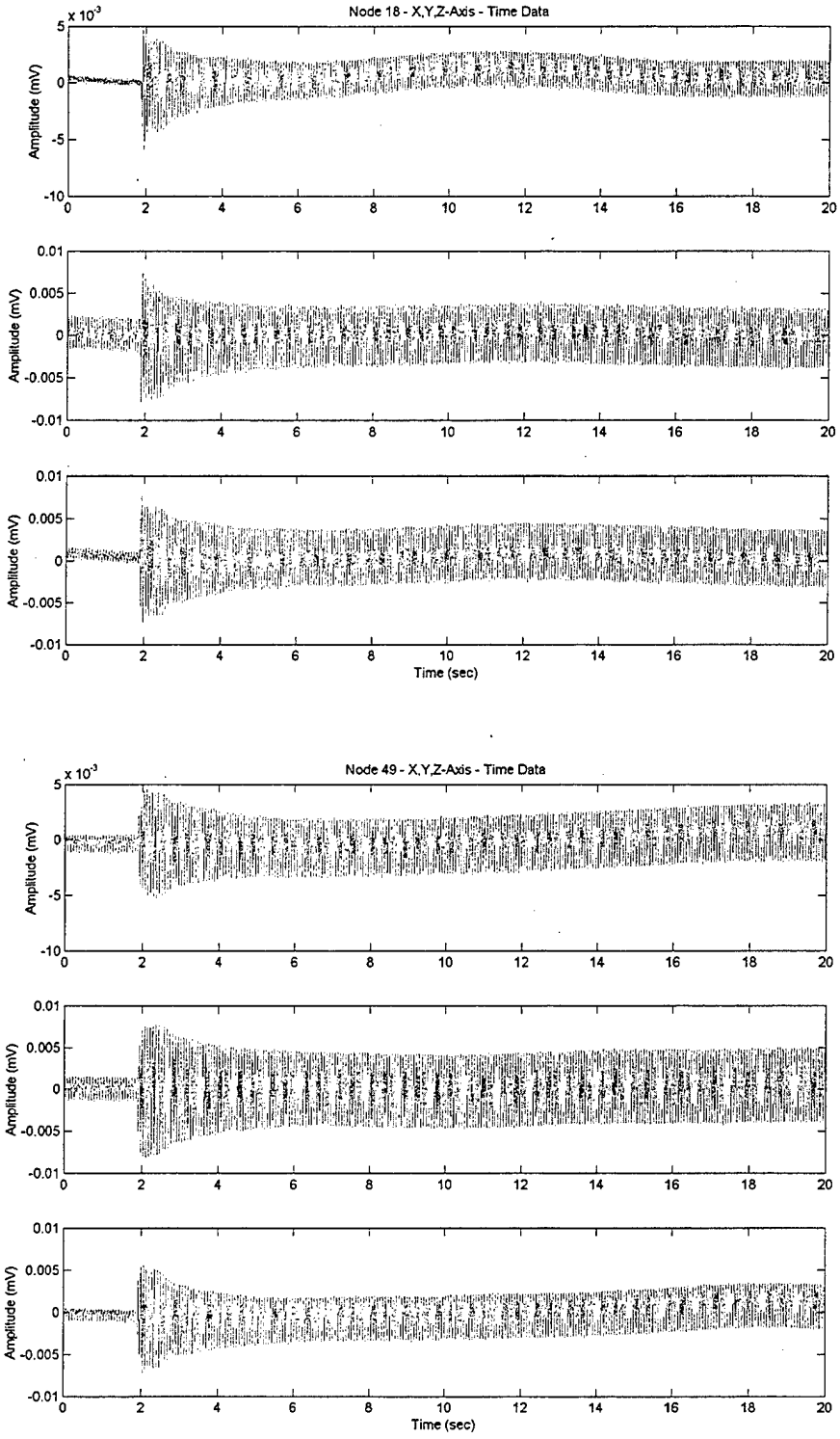
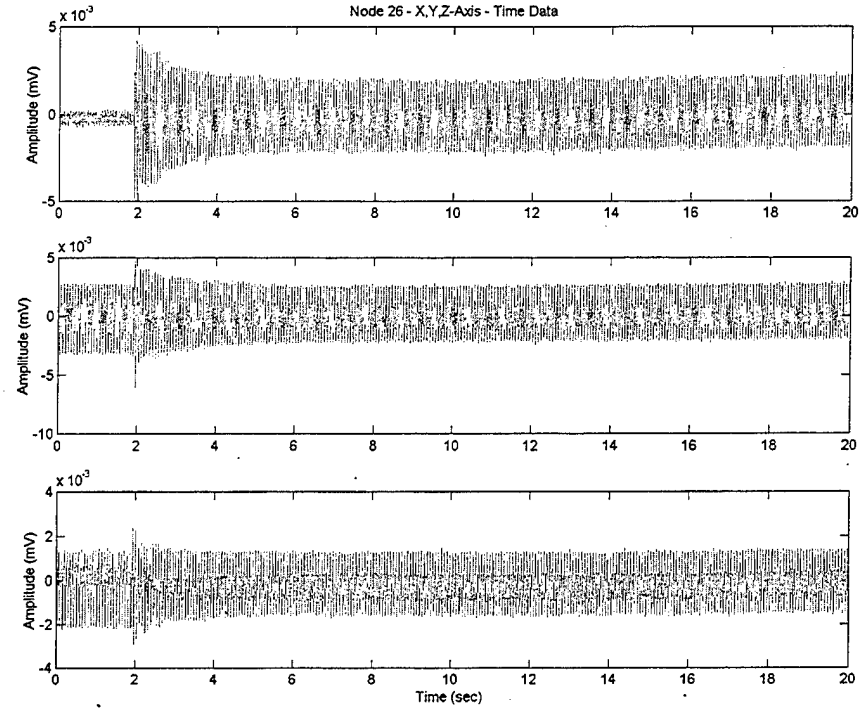


Figure 55. Exp. 2 Trial 8 Node 18 and 49 Response



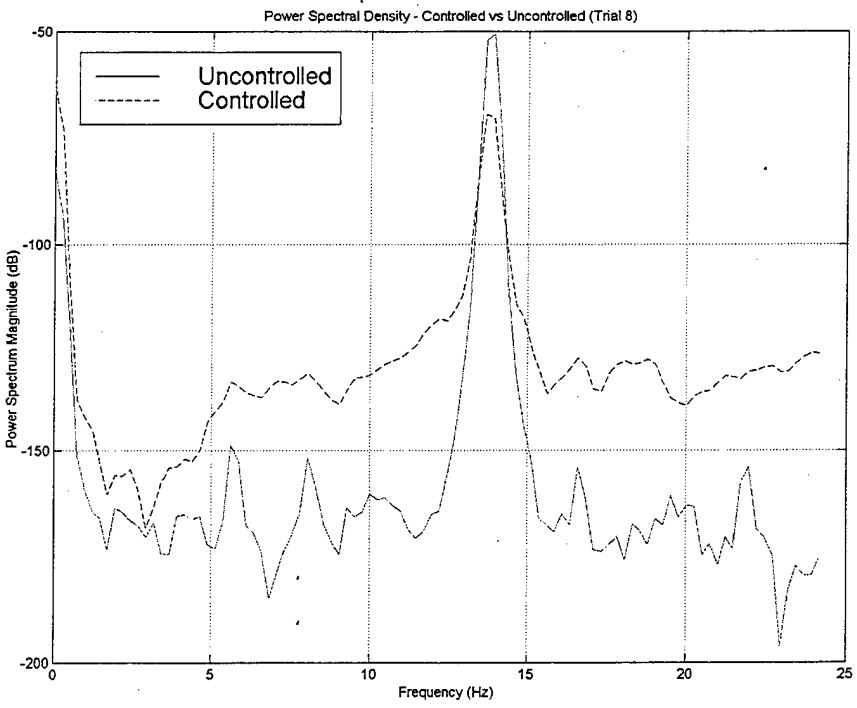


Figure 56. Exp. 2 Trial 9 Node 26 Response and PSD

EXPERIMENT 3 16.75 Hz 1.2 Vpp

Trial	IGain	IIGain	SG	Peak (dB)	Reduction	Comments	Plot
0	0	0	0	-52.7850	-	Uncontrolled	
1	300	0	2.00	-68.7235	15.9385	Borderline Unstable	
2	350	0	2.00	-71.1281	18.3432	Unstable	
3	350	0	1.75	-67.2985	14.5135	Slowly Unstable	
4	300	100	1.75	-70.6731	17.8882		
5	300	200	1.75	-70.4232	17.6382	Borderline Unstable	
6	0	-100	1.75	-57.2304	4.4455		
7	0	-200	1.75	-57.2934	4.5084	Noisy Output	
8	300	-100	1.75	-70.7148	**17.9299**	Best Case, This Trial	
9	300	-200	1.75	-70.7771	17.9921	Borderline Stable	
10	300	-300	1.75	-70.5887	17.8037		
11	300	-200	2.00	-74.3528	21.5678	Unstable	
12	0	-200	2.00	-57.2314	4.4465	Noisy	
13	0	1000	2.00	-57.1235	4.3385	Noisy	
14	0	-100	2.00	-56.2147	3.4298	Noisy	
15	0	0	0	-56.5548	-	Checks uncontrolled	
16	300	100	1.75	-69.7245	16.9396		

Table 21. Experiment 3 Results

EXPERIMENT 4 12.50 and 13.81 Hz, IGain=300, IIGain=100, SG=1.75

Trial	Phase	Peak (dB)	Reduction	Plot
10	0	-65.4948	-	
11	0	-72.7111	7.2163	
20	60	-65.4948	-	
21	60	-72.7861	7.2913	
30	120	-66.0130	-	
31	120	-72.5283	6.5153	
40	180	-66.1753	-	
41	180	-73.9794	7.8041	
50	240	-65.9113	-	
51	240	-75.1677	**9.2564**	**Y**
60	300	-66.3179	-	
61	300	-73.9508	7.6328	

Table 22. Experiment 4 Results

TRIAL 51: Phase = 240 degrees

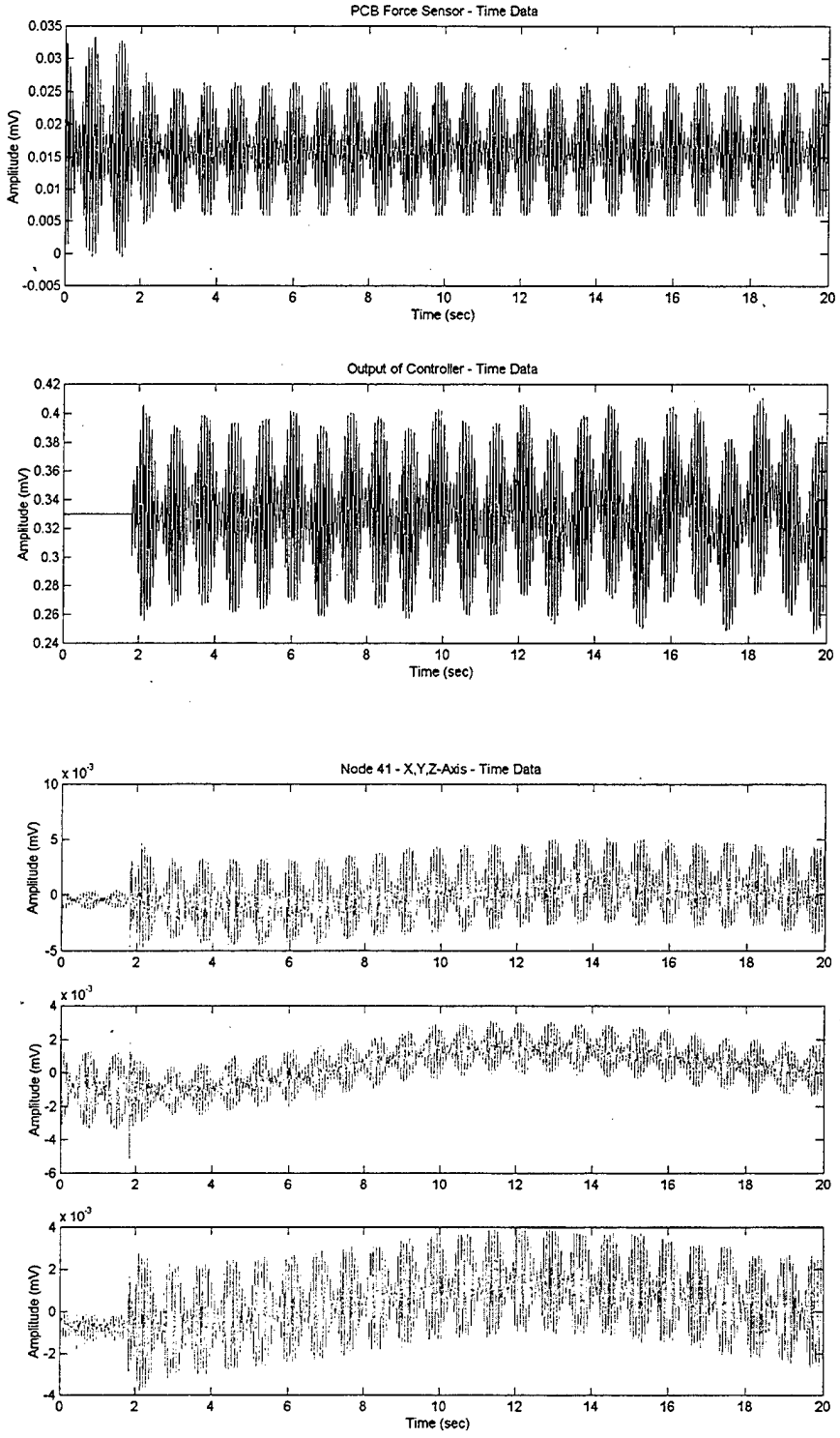


Figure 57. Exp. 4 Trial 51 Controller and Node 41 Response

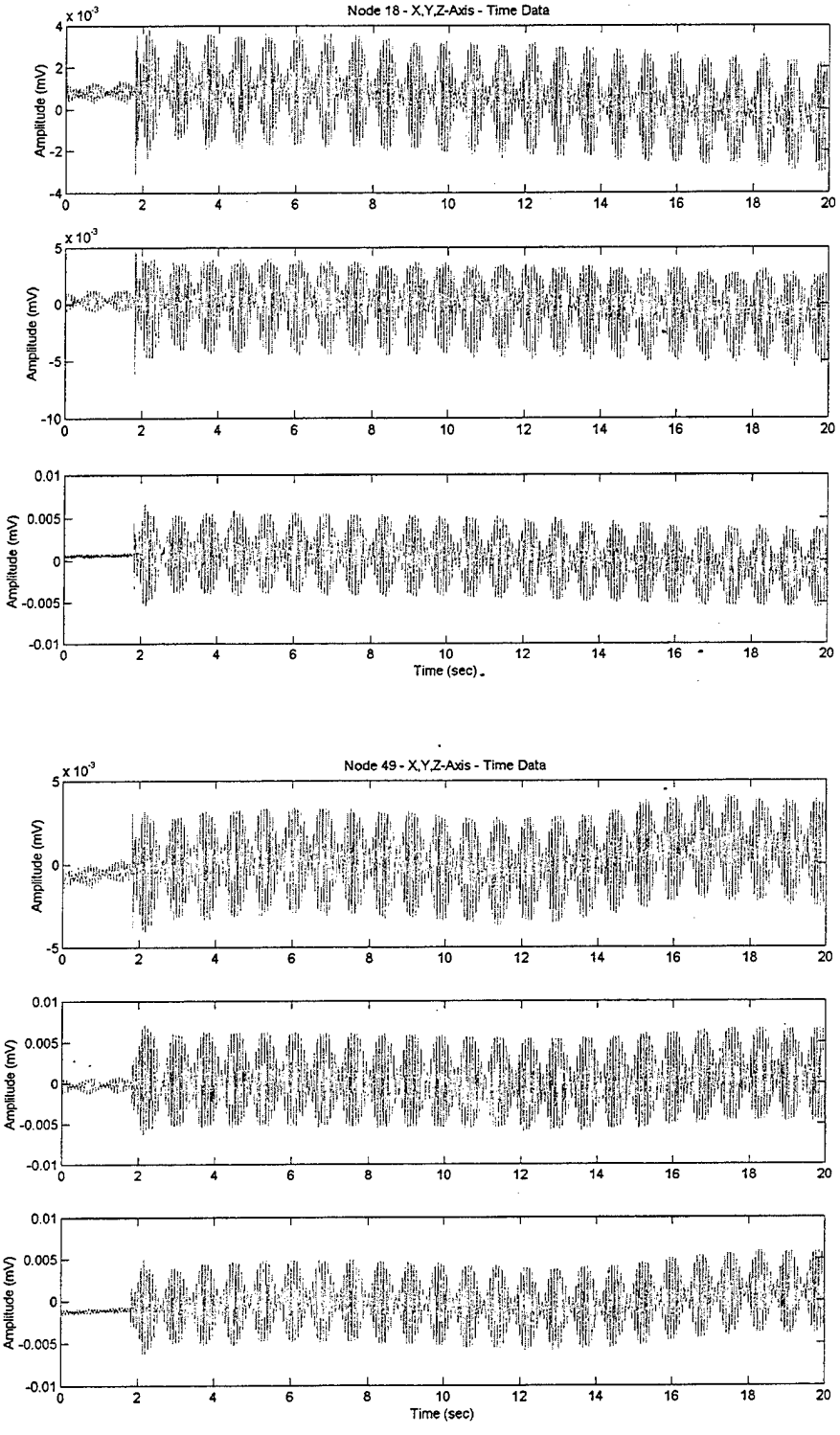
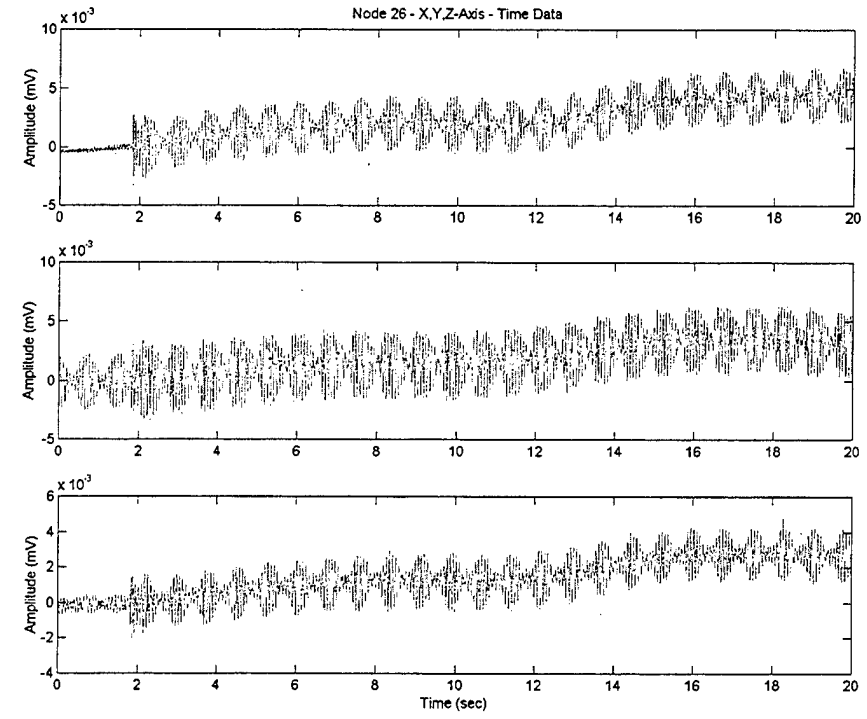


Figure 58. Exp. 4 Trial 51 Node 18 and 49 Response



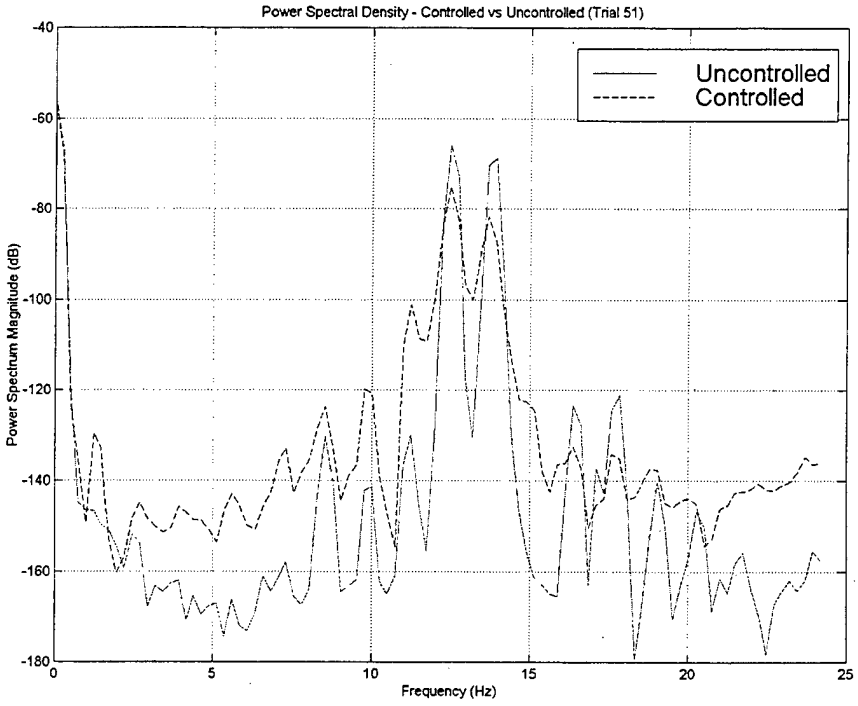


Figure 59. Exp. 4 Trial 52 Node 26 Response and PSD

EXPERIMENT 5 Random Noise 1.0 Vpp 10-35 Hz window
IGain=300, IIGain=100, SG=1.75

Trial	Uncontrolled	Trial	Controlled
1	-74.3642	11	-86.7251
2	-73.8561	12	-81.4073
3	-77.3457	13	-82.0824
4	-76.8609	14	-83.1579
5	-77.6162	15	-87.4192
6	-75.0826	16	-78.8793
7	-82.4348	17	-84.1822
8	-74.5222	18	-79.5973
9	-81.9090	19	-86.6130
10	-77.3291	20	-82.5681
Mean	-77.6503 +/- 5		-84.3384 +/- 5
	Change		6.6880

Table 23. Experiment 5 Results

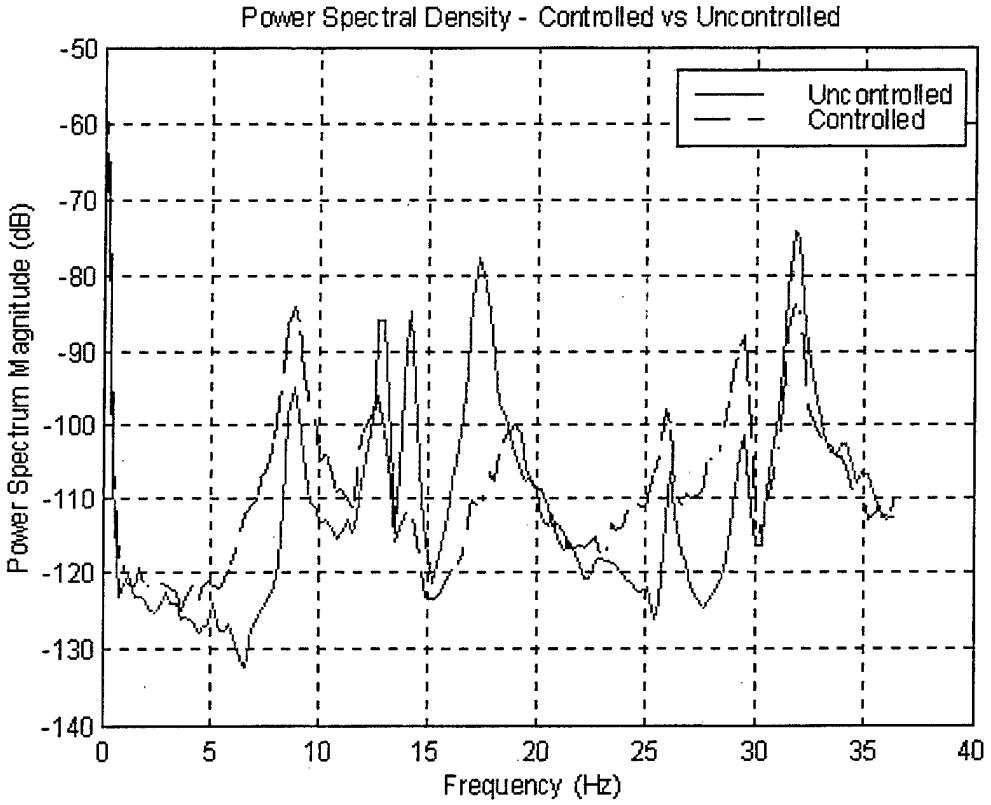


Figure 60. Exp. 5 Average Power Spectral Density

APPENDIX P. PROC_ANSYS.M

```
% ********************************************
% * Data Processing subroutine for ANSYS      *
% * Designed for the NPS Space Truss Experiment *
% * By LT Carey M. Pantling, USN              *
% * Last Modified 10/21/99                    *
% ********************************************

freq=11.75;                          % driving frequency
FS=20*freq;                          % sampling frequency
NS=940;                              % number of samples
time=[1/FS:1/FS:NS/FS];              % time vector
expt=1;                              % experiment number
% AN - node 26-y output
% AR - output to piezo
% AS - Input from force transducer

for n = 0:28                         % runs through all the data
   % load the data and get into a good name
   if n<10                           % gives leading zero
      eval(['load AN_e',num2str(expt),'_0',num2str(n),'.out;']);
      eval(['AN = AN_e',num2str(expt),'_0',num2str(n),';']);
      eval(['load AR_e',num2str(expt),'_0',num2str(n),'.out;']);
      eval(['AR = AR_e',num2str(expt),'_0',num2str(n),';']);
      eval(['load AS_e',num2str(expt),'_0',num2str(n),'.out;']);
      eval(['AS = AS_e',num2str(expt),'_0',num2str(n),';']);
   else                              % no leading zero
      eval(['load AN_e',num2str(expt),'_',num2str(n),'.out;']);
      eval(['AN = AN_e',num2str(expt),'_',num2str(n),';']);
      eval(['load AR_e',num2str(expt),'_',num2str(n),'.out;']);
      eval(['AR = AR_e',num2str(expt),'_',num2str(n),';']);
      eval(['load AS_e',num2str(expt),'_',num2str(n),'.out;']);
      eval(['AS = AS_e',num2str(expt),'_',num2str(n),';']);
   end
   figure(1);                        % plot the raw data
         subplot(3,1,1);
            plot(time,AS);
            title('PCB Force Sensor and Controller Output')
         ylabel('Amplitude (V)');
         axis([0 4 -1 1]);
      subplot(3,1,2);
            plot(time,AR);
         ylabel('(V)');
         axis([0 4 -80 80]);
      subplot(3,1,3);
            plot(time,AN);
               title('Node 26-y Motion')
         ylabel('(meters)');
            xlabel('Time (sec)');
      eval(['print -djpeg t',num2str(n),'fig1.jpg;']);
   figure(2);                        % plot the processed data
   [Pxx1,F1]=PSD(AN,512,FS);         % creates PSD of node 26-Y
```

```
    if n==0                              % plot only the uncontrolled PSD
          plot(F1(1:100),20*log10(Pxx1(1:100)));
          title(['Power Spectral Density - Uncontrolled']);
       ylabel('Power Spectrum Magnitude (dB)');
          xlabel('Frequency (Hz)');
       legend('Uncontrolled',0);
       axis([0 40 -300 -100]);
          grid;
       Pxx2=Pxx1;
       save Pxx2 Pxx2;                   % save the uncontrolled PSD
    else                                 % evaluate the controlled cases
       load Pxx2;                        % load uncontrolled case
       plot(F1(1:100),20*log10(Pxx2(1:100)),'r-',F1(1:100), ...
          20*log10(Pxx1(1:100)),'b--')
       title(['Power Spectral Density - Controlled vs Uncontrolled', ...
          ' - Trial ',num2str(n),')']);
          ylabel('Power Spectrum Magnitude (dB)');
          xlabel('Frequency (Hz)');
       legend('Uncontrolled','Controlled',0);
       axis([0 40 -300 -100]);
       grid;
       eval(['print -djpeg t',num2str(n),'fig2.jpg;']);
       u = max(20*log10(Pxx2(10:100)));
          c = max(20*log10(Pxx1(10:100)));
          diff = u - c;                  % dB reduction due to control
       results(n,:)=[n u c diff]
    end                                  % of plot routine
end                                      % of loop for trial info
results                                  % spit results to screen
```

APPENDIX Q. ANSYS RESULTS

Series 1: 11.30 Hz

Trial	IGain	IIGain	SG	Peak (dB)	Reduction	Comments	Plot
0	0	0	0	-103.5685	--	Uncontrolled Case	
1	300	0	-20	-111.0516	7.4831		
2	300	0	-40	-113.5936	10.0251		
3	300	0	-50	-114.1892	10.6207		
4	300	0	-75	-114.9408	11.3722		
5	300	0	-100	-115.2221	11.6536		
6	300	0	-125	-115.3466	11.7781	Close to Saturated	
7	300	0	-150	-115.3658	11.7973	Saturation Limit	
8	300	0	-175	-115.4091	11.8406	Saturated	
9	300	0	-200	-115.4285	11.8599	Saturated IGain	**Y**
10	0	50k	-150	-127.0601	23.4915	Saturated IIGain	**Y**
11	0	25k	-150	-121.8247	18.2562	Saturated	
12	0	20k	-150	-123.5433	19.9748	Saturated	
13	0	15k	-150	-125.1943	21.6258	Saturated	
14	0	10k	-150	-128.5719	25.0034	Saturated	
15	0	8k	-150	-129.6683	26.0998	Slightly Saturated	**Y**
16	0	5k	-150	-126.0206	**22.4521**	**Best Overall Case**	**Y**
17	0	8k	-75	-122.0761	18.5076		
18	0	5k	-75	-113.4415	9.8730		
19	0	2k	-75	-105.9514	2.3829		
20	300	200	-75	-114.9759	11.4074		
21	300	500	-75	-115.0428	11.4743		
22	300	1000	-75	-115.1485	11.5800		
23	300	2k	-75	-105.9514	2.3829	Questionable	
24	300	4k	-75	-110.5684	6.9999	Questionable	
25	300	8k	-75	-116.6247	13.0562		
26	300	10k	-75	-116.9850	13.4165		
27	300	15k	-75	-117.6994	14.1309		

Table 24. ANSYS Series 1 Results

Trial 9: IGain=300, IIGain=0, SG= -200.

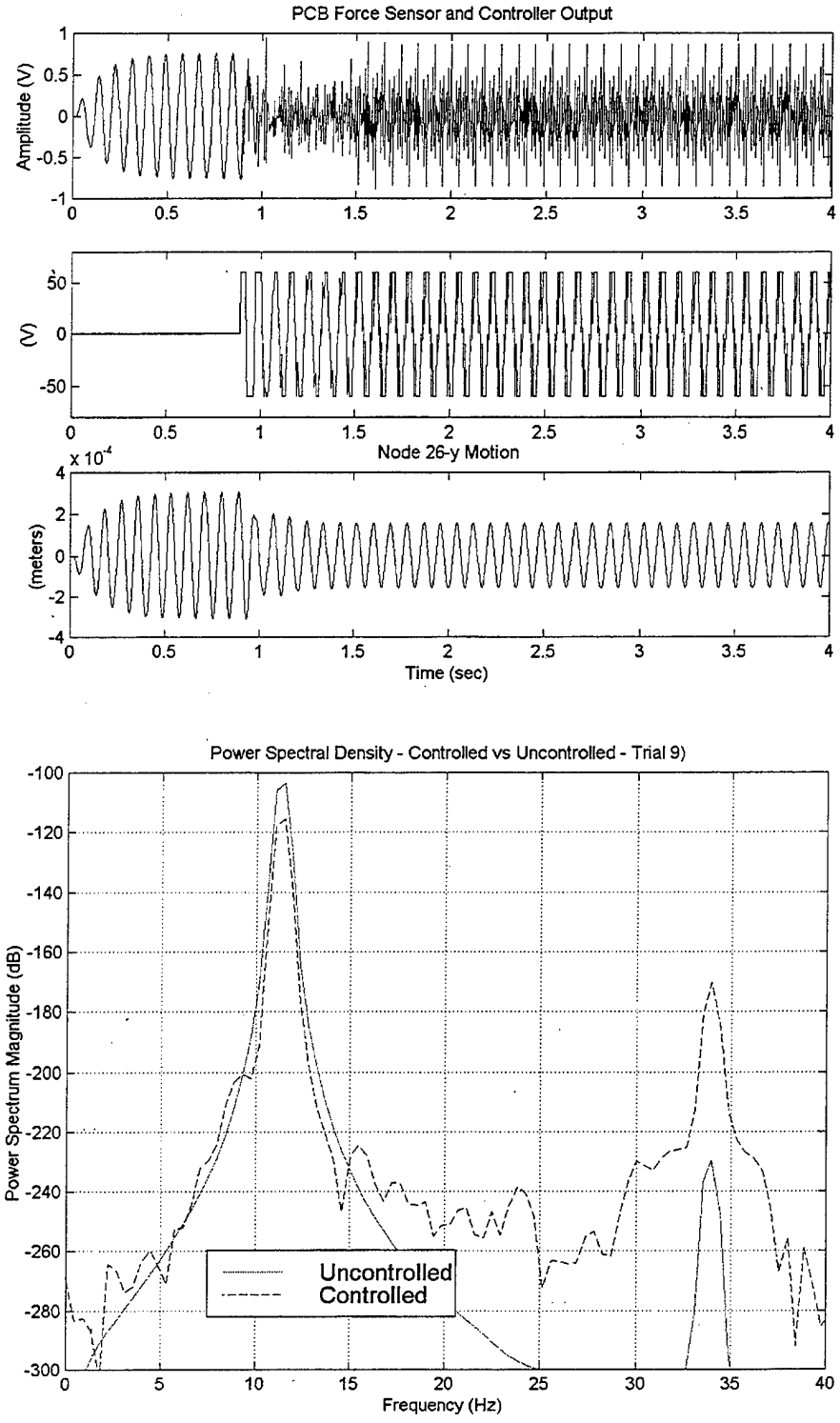
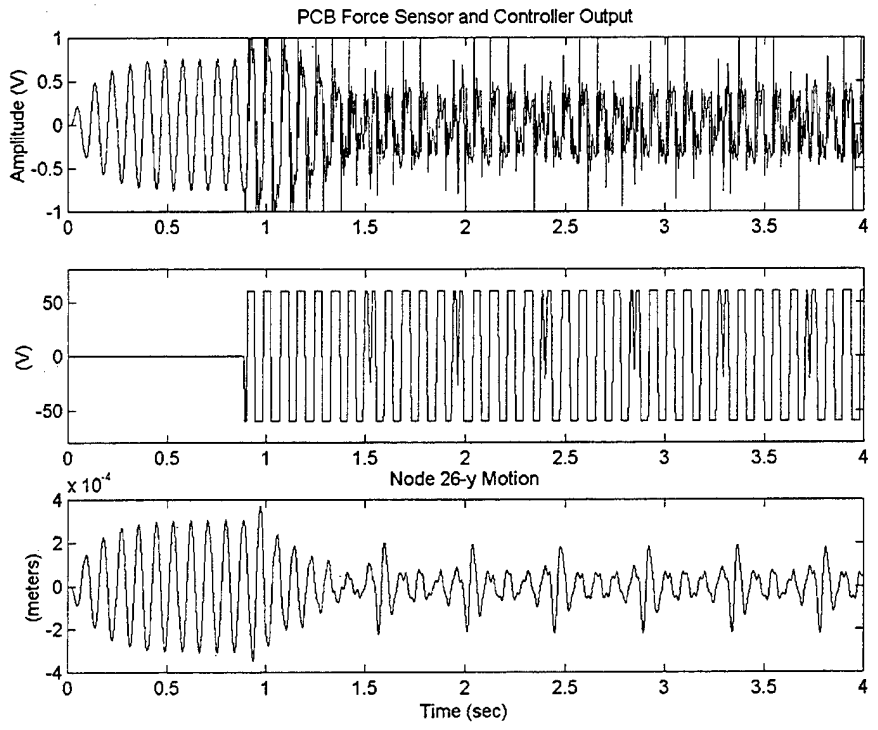


Figure 61. Series 1 Trial 9 Results and PSD

Trial 10: IGain=0, IIGain=50,000, SG= -150.



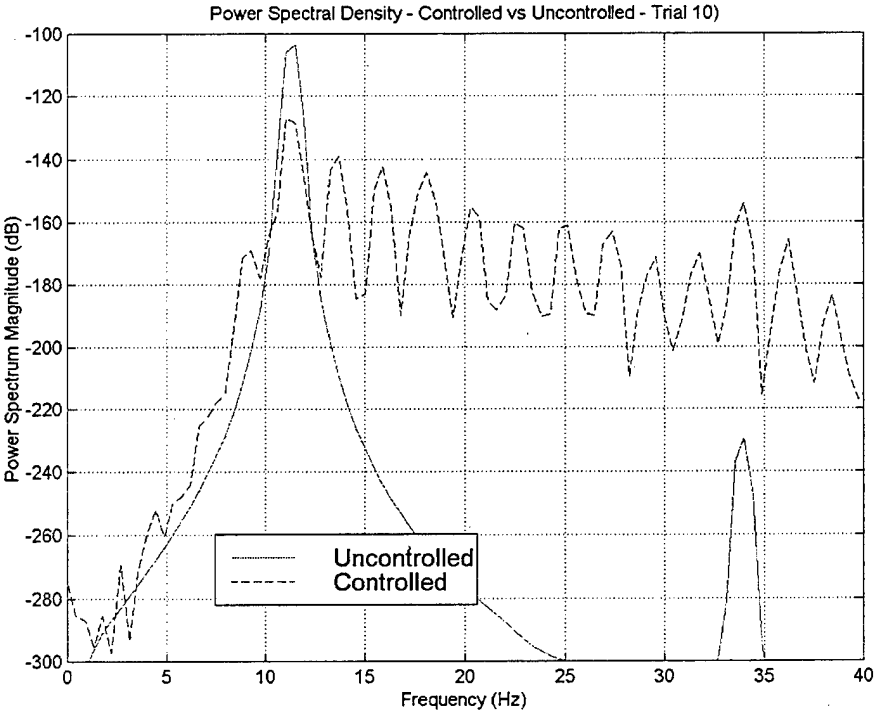


Figure 62. Series 1 Trial 10 Results and PSD

Trial 15: IGain=0, IIGain=8,000, SG= -150.

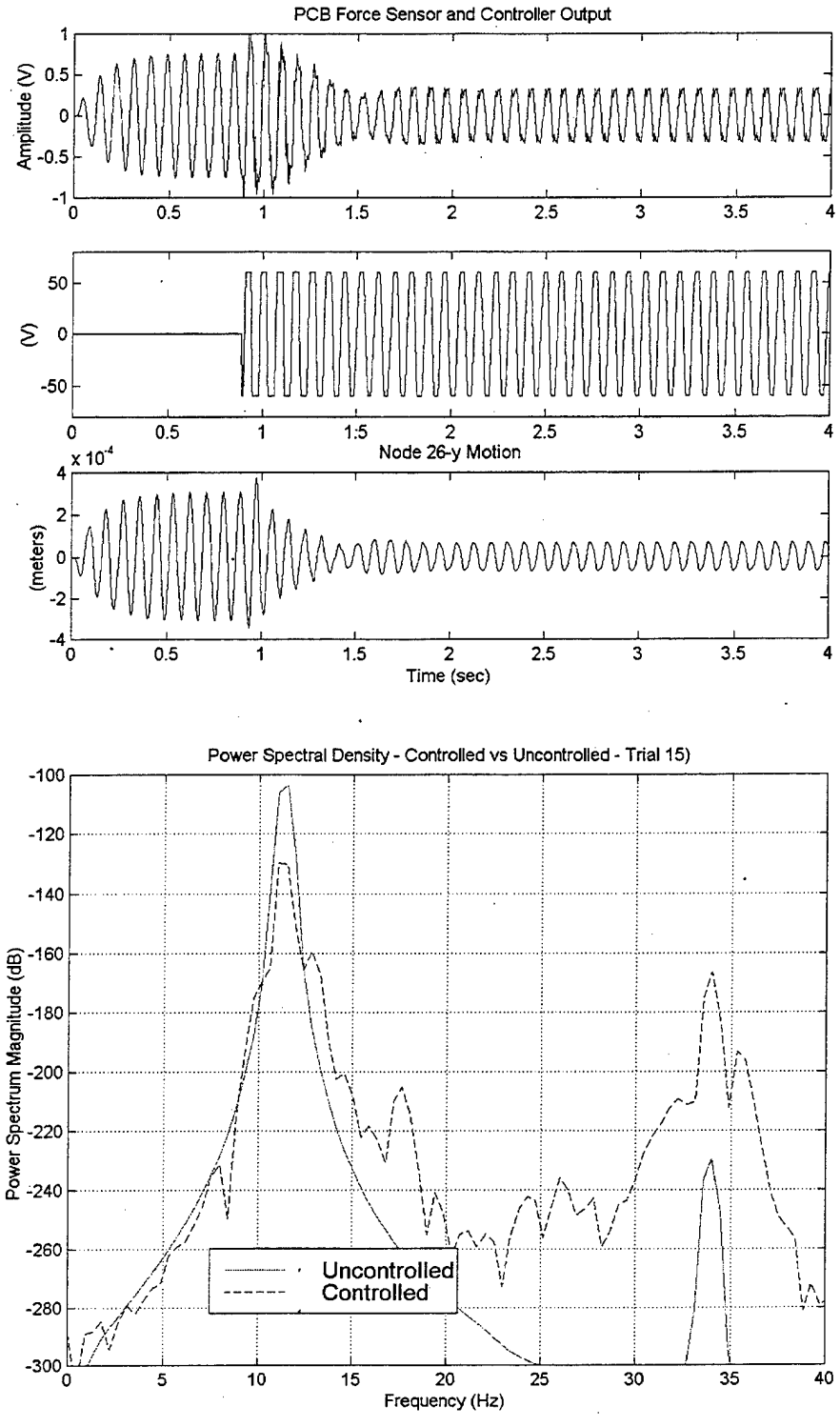
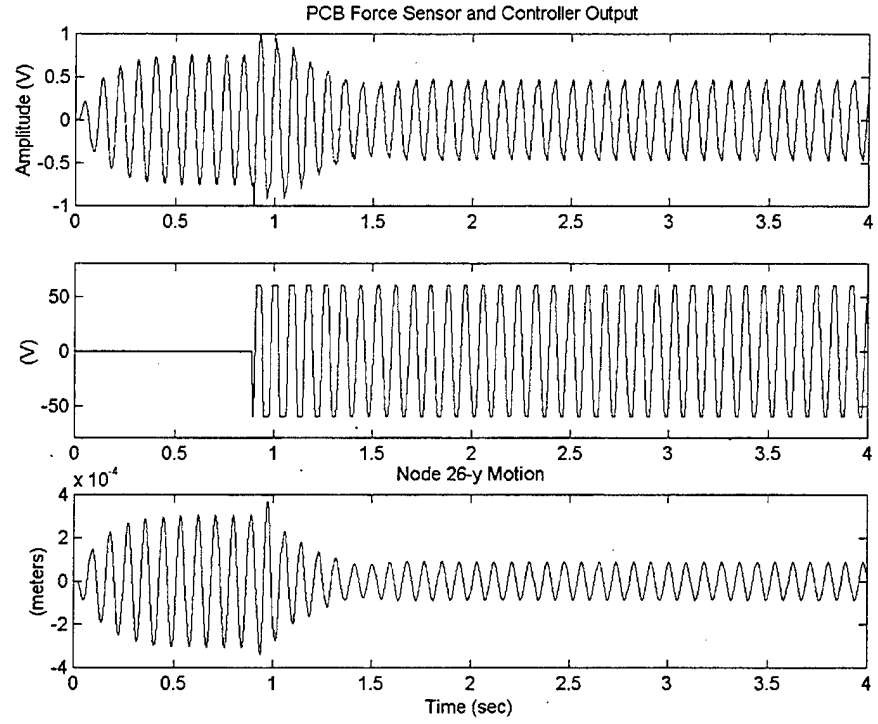


Figure 63. Series 1 Trial 15 Results and PSD

Trial 16: IGain=300, IIGain=5,000, SG= -150.



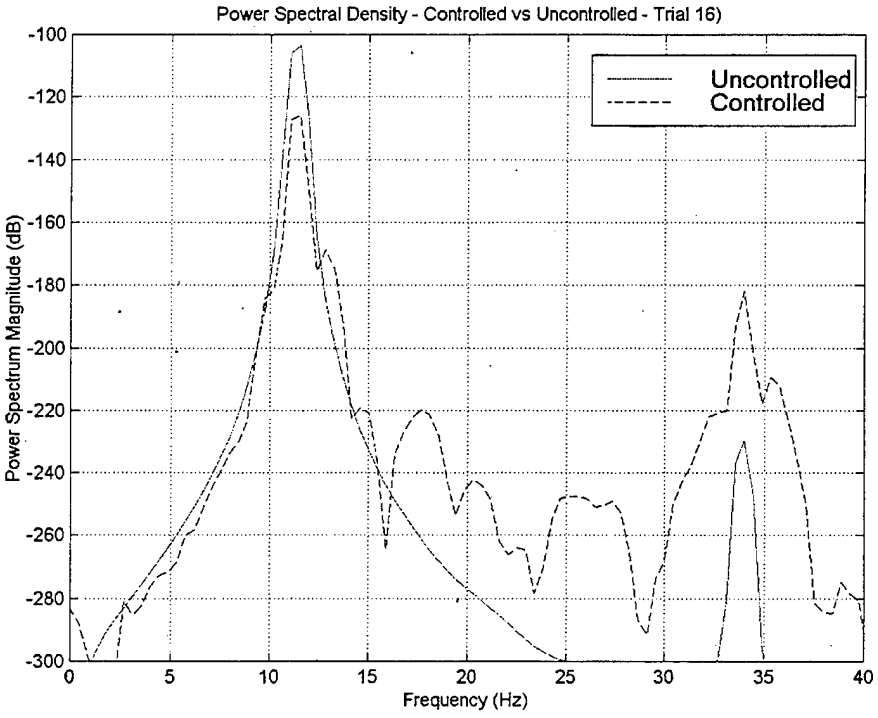


Figure 64. Series 1 Trial 16 Results and PSD

Series 2: 11.75 Hz

Trial	IGain	IIGain	SG	Peak (dB)	Reduction	Comments	Plot
0	0	0	0	-104.6728	--	Uncontrolled Case	
1	300	0	-20	-112.8698	8.1971		
2	300	0	-40	-116.3466	11.6738		
3	300	0	-50	-117.2903	12.6175		
4	300	0	-75	-118.6112	13.9385		
5	300	0	-100	-119.2355	14.5627		
6	300	0	-125	-119.5663	14.8936	Saturated Limit	
7	300	0	-150	-119.6658	14.9930	Saturated	
8	300	0	-175	-119.7329	15.0601	Saturated	
9	300	0	-200	-119.6301	14.9573	Saturated	
10	0	50k	-150	-128.1796	23.5068	Saturated	
11	0	25k	-150	-125.7621	21.0893	Saturated	
12	0	20k	-150	-126.0112	21.3384	Saturated	
13	0	15k	-150	-126.6791	22.0063	Saturated	
14	0	10k	-150	-128.9777	24.3049	Saturated	
15	0	8k	-150	-129.6968	25.0240	Saturated	
16	0	5k	-150	-123.0963	18.4235	Saturated	
17	0	8k	-75	-118.3636	13.6908	Unsaturated	
18	0	5k	-75	-110.1677	5.4949		
19	0	2k	-75	-104.9327	0.2599		
20	300	200	-100	-119.2771	14.6043		
21	300	500	-100	-119.3445	14.6717		
22	300	1000	-100	-104.6185	0.0543	Questionable	
23	300	2k	-100	-119.6143	14.941		
24	300	4k	-100	-120.0087	15.336		
25	300	6k	-100	-120.3793	15.7065		
26	300	8k	-100	-120.7338	16.0611		
27	300	10k	-100	-121.0980	16.4253		
28	300	12k	-100	-121.3796	16.7068		
29	300	15k	-100	-121.7479	17.0752		
30	300	20k	-100	-122.2401	17.5674		
31	300	25k	-100	-122.4973	17.8245		
32	300	30k	-100	-122.5947	17.9219		
33	300	35k	-100	-122.6512	17.9784		
34	300	40k	-100	-122.7332	**18.0605**	**Best Case This Series**	

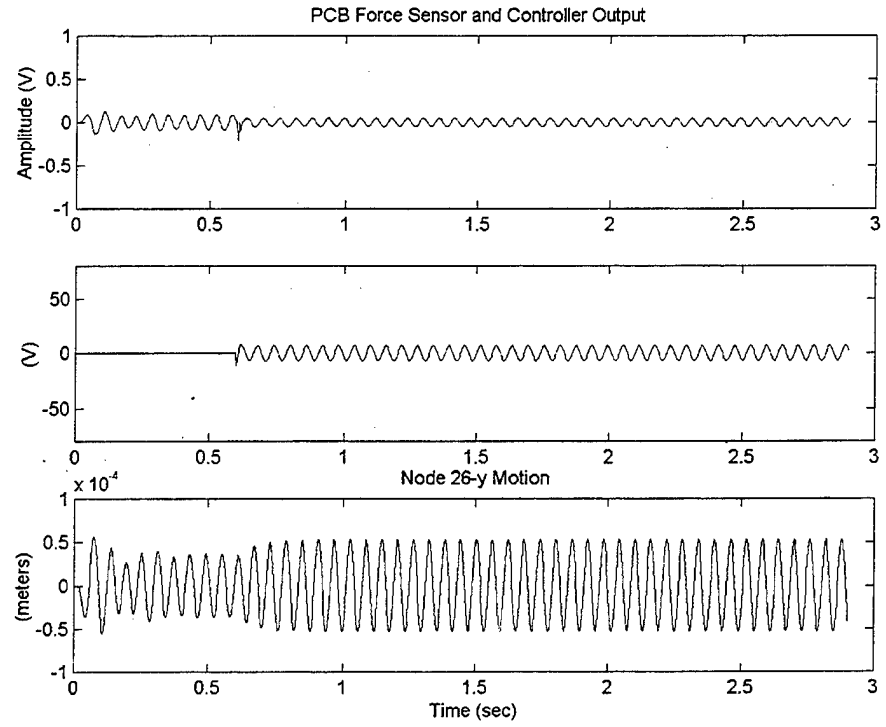
Table 25. ANSYS Series 2 Results

Series 3: 16.75 Hz

Trial	IGain	IIGain	SG	Peak (dB)	Reduction	Comments	Plot
0	0	0	0	-140.6223	--	Uncontrolled Case	
1	300	0	-20	-136.3002	-4.3220	Amplifies Signal	
2	300	0	-40	-134.5450	-6.0772	Amplifies Signal	
3	300	0	-50	-134.0553	-6.5670	Amplifies Signal	**Y**
4	300	0	25	-160.2686	19.6464	Starting to Destabilize	**Y**
5	300	0	50	-142.6854	2.0631	Negligible Effect	**Y**
6	300	0	75	-140.5569	-0.0654	Unstable to (-) limit	
7	300	0	100	-140.9973	0.3750	Unstable	
8	300	0	125	-141.0681	0.4459	Unstable	
9	300	0	150	-141.1044	0.4821	Unstable	
10	300	0	175	-140.8921	0.2698	Unstable	
11	300	0	200	-141.1225	0.5003	Unstable	
12	0	20k	150	-140.7954	0.1731	Unstable	
13	0	15k	150	-140.5506	0.0717	Unstable	
14	0	10k	150	-140.5827	0.0395	Unstable	
15	0	5k	150	-141.0630	0.4408	Unstable	
16	0	5k	75	-141.4760	0.8537	Unstable	
17	0	2k	75	-141.4851	0.8628	Starting to Destabilize	
18	0	1k	75	-141.1150	0.4928	Negligible Effect	
19	0	500	75	-140.8755	0.2533	Negligible Effect	

Table 26. ANSYS Series 3 Results

Trial 3: IGain=300, IIGain=0, SG= -50.



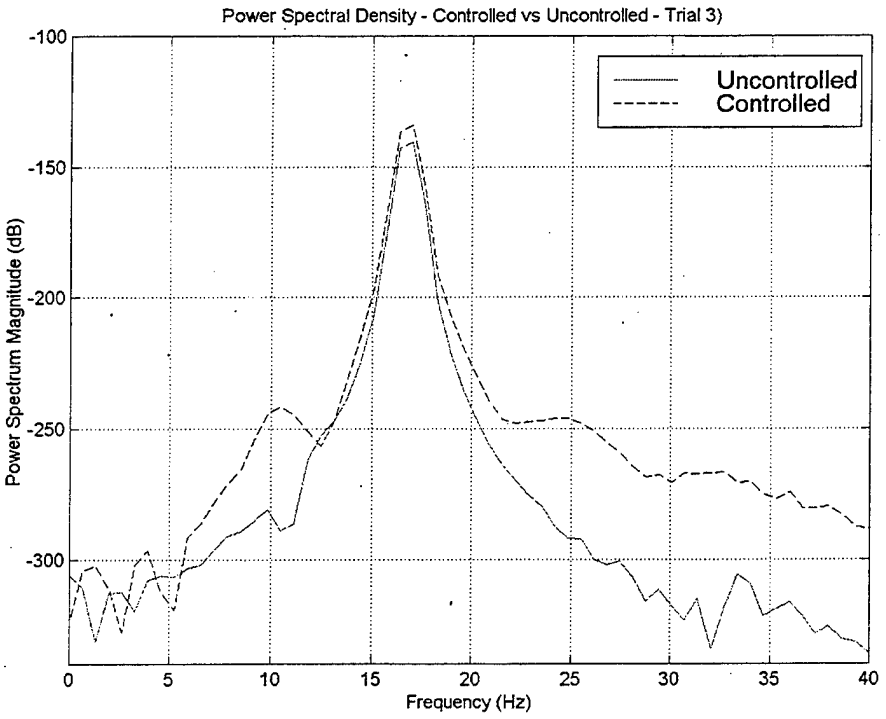


Figure 65. Series 3 Trial 3 Results and PSD

Trial 4: IGain=300, IIGain=0, SG= 25.

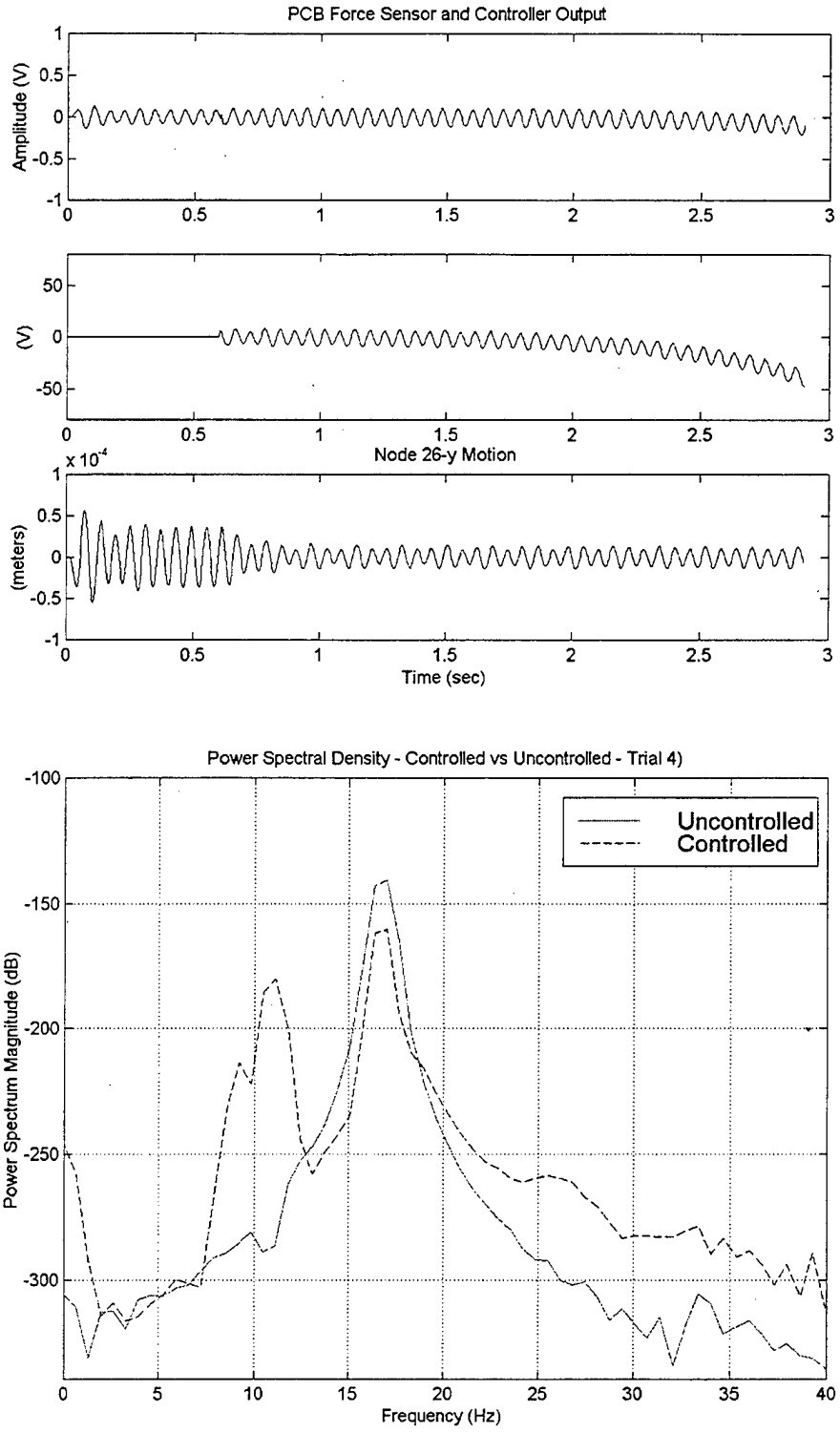
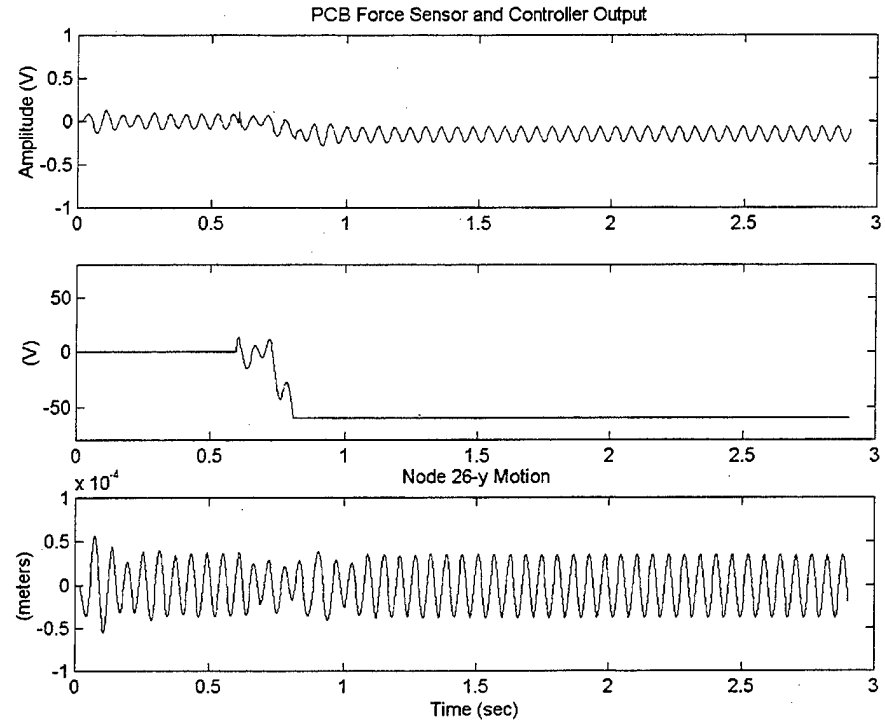


Figure 66. Series 3 Trial 4 Results and PSD

Trial 5: IGain=300, IIGain=0, SG= 50.



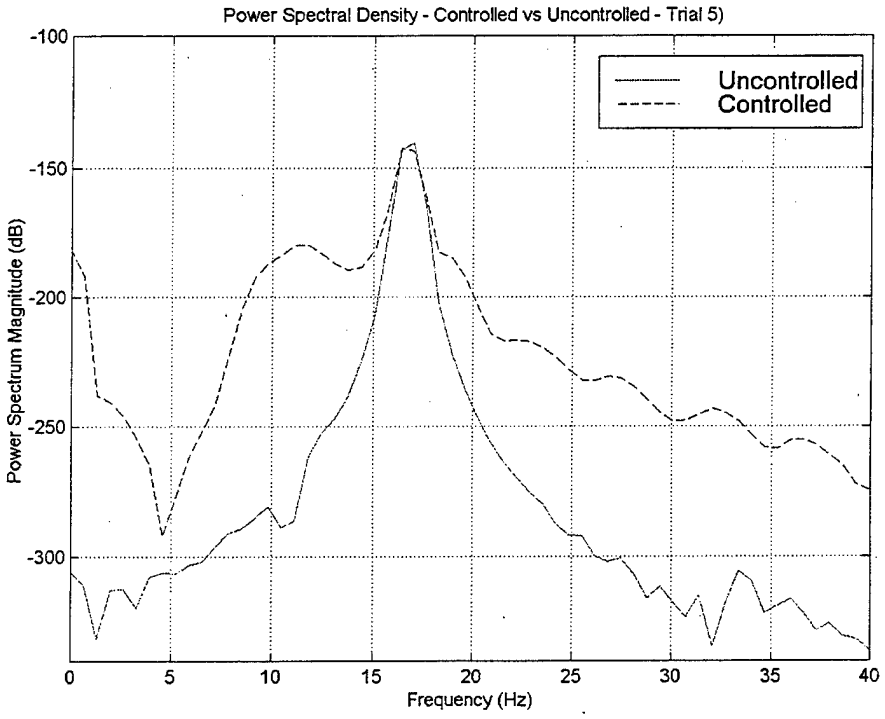


Figure 67. Series 3 Trial 5 Results and PSD

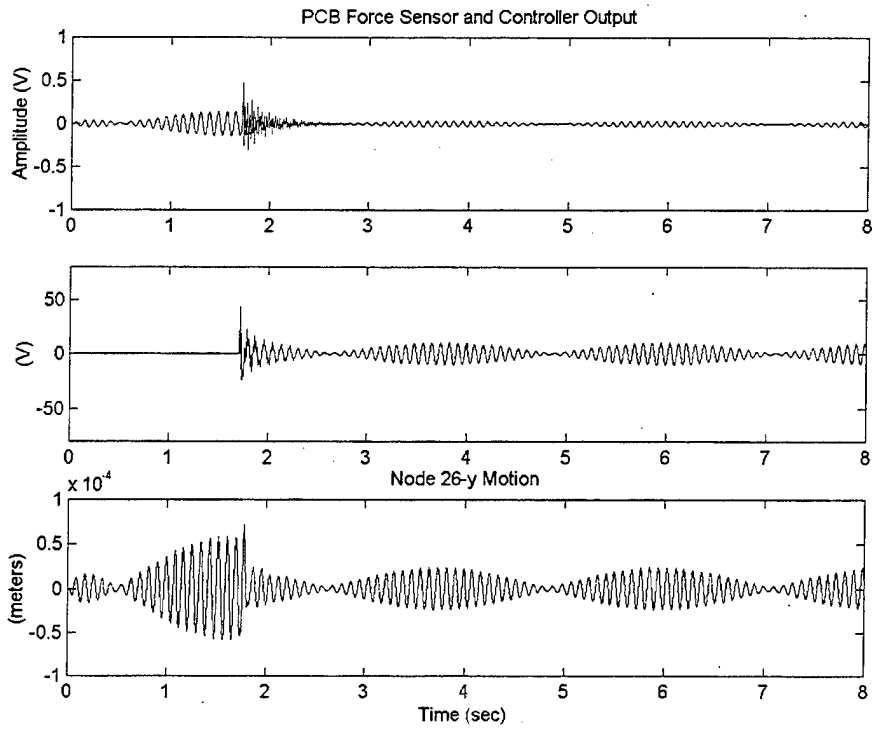
Series 4: 11.30 and 11.75 Hz

IGain=300, IIGain=0, SG= -75

Trial	Phase	Peak (dB)	Reduction	Comments	Plot
10	0	-141.8379	--		
11	0	-156.9130	15.0751		
20	60	-142.1832	--		
21	60	-157.2118	15.0285		
30	120	-141.6837	--		
31	120	-157.2496	15.5659		Y
40	180	-142.1409	--		
41	180	-156.9814	14.8405		
50	240	-141.7964	--		
51	240	-156.6923	14.8959		
60	300	-141.6455	--		
61	300	-156.6559	15.0104		

Table 27. ANSYS Series 4 Results

Trial 4: IGain=300, IIGain=0, SG= -50 Phase =120 degrees.



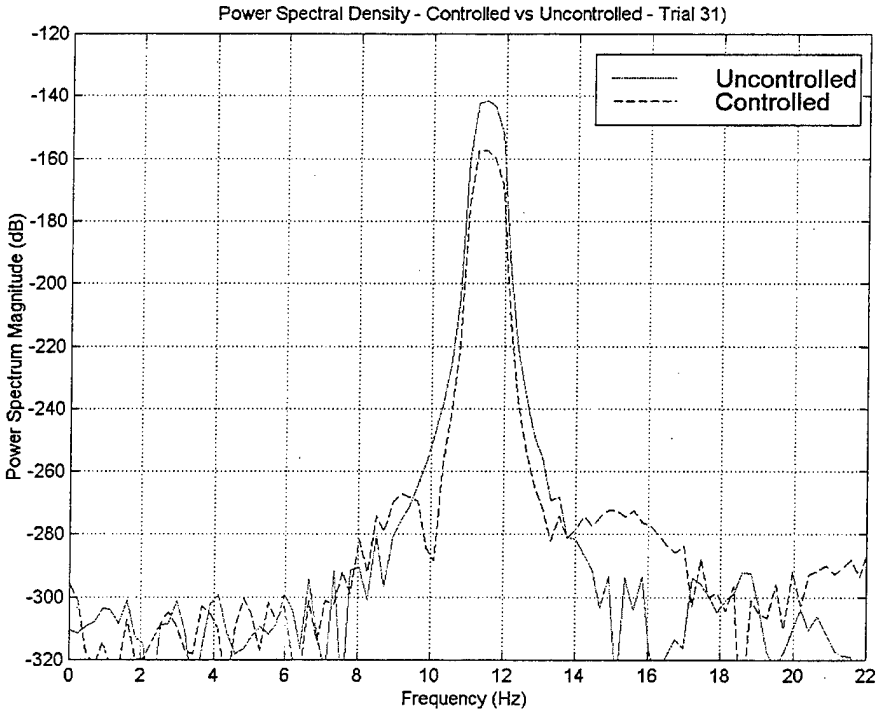


Figure 68. Series 4 Trial 31 Results and PSD

Series 5: Random Noise (0-200Hz)

IGain=300, IIGain=0, SG= -75

Peak Uncontrolled Amplitude = -199.2025dB
Peak Controlled Amplitude = -205.2396dB
Reduction = 6.0371dB

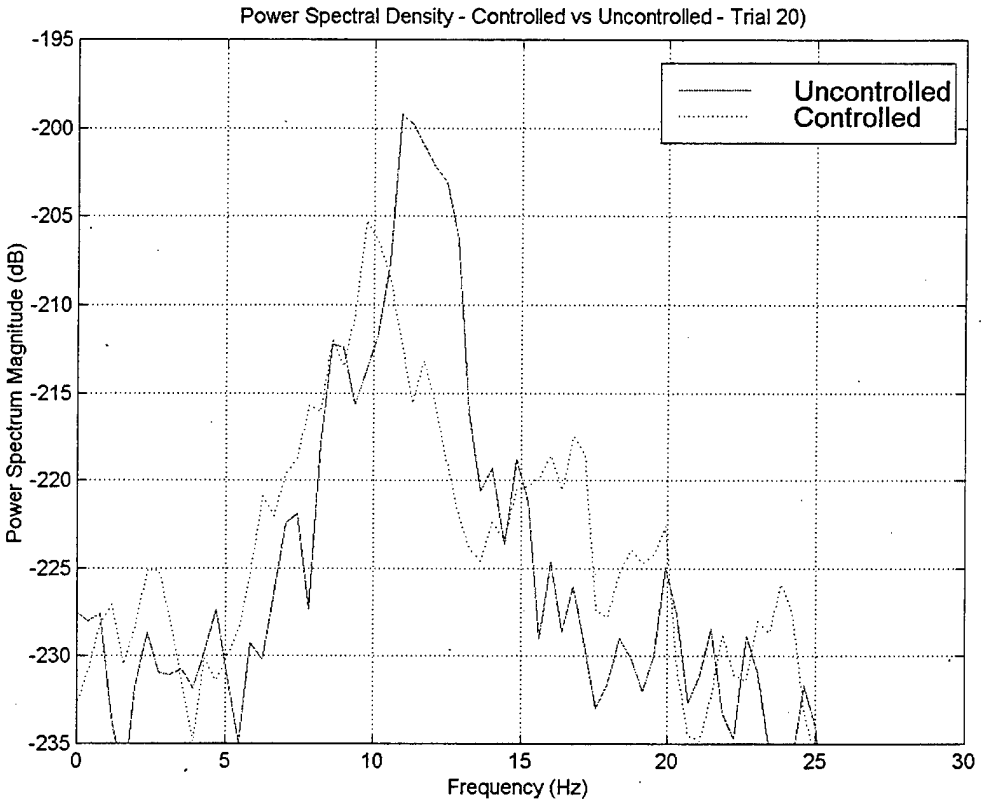


Figure 69. Series 5: Random Power Spectral Density

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvior, Virginia 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, California 93943-5101

3. LT Carey M. Pantling, USN 1
   4 Sherwood Drive
   Westerly, Rhode Island 02891

4. Professor Young S. Shin Code ME/Sg 2
   Naval Postgraduate School
   699 Dyer Road Room 137
   Monterey, California 93943-5106

5. Professor Brij N. Agrawal Code AA/Ag 2
   Naval Postgraduate School
   699 Dyer Road Room 137
   Monterey, California 93943-5106

6. Sheldon Imaoka 1
   Collaborative Solutions, Inc.
   2303 W. 190th Street
   Redondo Beach, California 90278

7. Dr. Shalom Fisher 1
   Naval Research Laboratory Code 8220
   4555 Overlook Ave SW
   Washington, District of Columbia 20375

8. Department of Aeronautics and Astronautics Code AA 1
   Naval Postgraduate School
   699 Dyer Road Room 137
   Monterey, California 93943-5106

9. Dr. Charles Daniel Code SP/CD 1
   Naval Postgraduate School
   699 Dyer Road Room 137
   Monterey, California 93943-5106

10. Joseph Dutson 1
MC ZC01
502 Gemini Street
Houston, Texas, 77058

11. OB/Victor M. Cooley 1
Johnson Space Center
Houston, Texas 77058